DataBoard

1004-10

BESKRIVNING

Z80A-CTC-SIO SINGLE-BOARD-COMPUTER

1004-10(B)

Sep 84

DATA INDUSTRIER AB
Box 2029, 18302 TÄBY. Tel 08-768 06 60. Telex 10978.

CONTENTS

INNEHÅLL

DESCRIPTION

The 1004-10 is a Single Board computer, based on Z80A-CPU, Z80A-CTC and Z80A-SIO/2. It contains memory sockets for up to 24 Kbytes EPROM or up to 16 Kbytes EPROM and up to 8 Kbytes BYTEWYDE static RAM.

The CTC and SIO provides 4 counters and 2 channels of serial data communication. In addition a "1 of 8" code input, activated by a strobe, is available.

The 1004 is delivered with 2.5 MHz system clock, according to the DataBoard 4680 standard, but can be modified for 4 MHz by changing the crystal.

Expansion of the memory and I/O is through the DataBoard single board computer bus. With 4 MHz, the access time limitations must be considered.

The 1004 does not support DMA or dynamic memories. The double board computer 1045/1046 should be used when these features or more then 2 interrupt levels are required. (Compare 4088.)

BESKRIVNING

1004-10 är en enkortsdator, baserad på Z80A-CPU, Z80A-CTC och Z80A-SIO/2. Den innehåller minnessocklar för upp till 24 Kbytes EPROM eller upp till 16 Kbytes EPROM och upp till 8 Kbytes BYTEWYDE statiskt RAM.

CTC och SIO innehåller 4 räknare och 2 kanaler för seriell datakommunikation. Dessutom finns en "1 av 8" kod-ingång, aktiverad av en strobe.

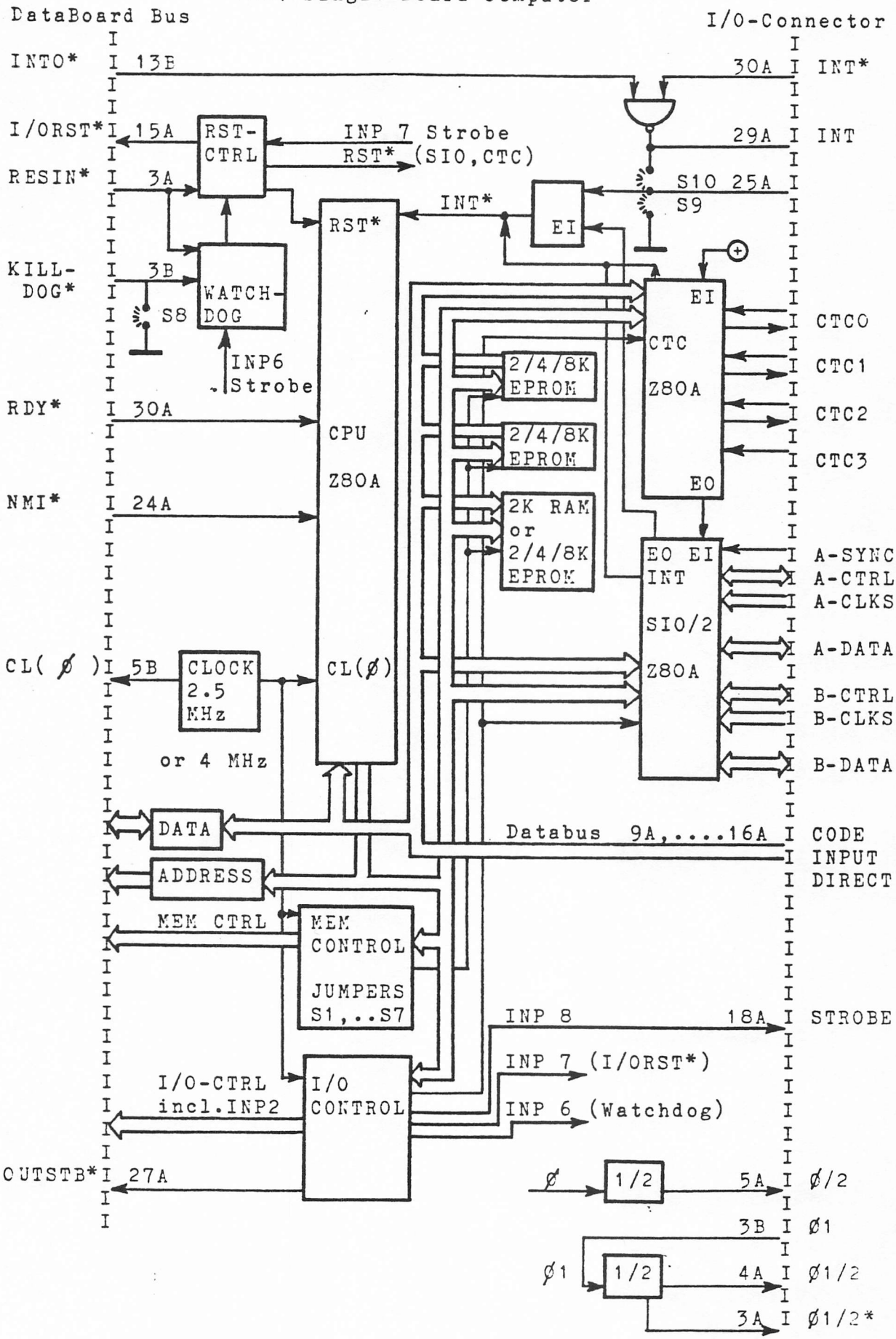
1004 levereras med 2.5 MHz systemklocka, enligt DataBoard 4680 standard, men kan modifieras för 4 MHz genom att byta kristall.

Expansion av minne och I/O sker via DataBoard enkorts-datorbuss. Då 4 MHz används, måste accesstidsbegränsningarna beaktas.

1004 kan ej hantera DMA eller dynamiska minnen. Tvåkorts-datorn 1045/1046 bör användas när denna funktion eller mer är 2 interruptnivåer behövs. (Jämför 4088).

BLOCK DIAGRAM BLOCK SCHEMA

1004 Single Board Computer



TECHNICAL DATA	TEKNISKA DATA
Processor	Z80A (4 MHz version)
Number of instructions:	158, includes 4, 8 and 16 bit operations.
Antal instruktioner:	158, inkl. 4, 8 och 16 bits operationer.
System clock:	-Standard 2.5 MHz. Optional 4 MHz by replacing the 5 MHz crystal with an 8 MHz crystal, type HC18 serie resonant.
Systemklocka:	-Standard 2.5 MHz. Valbart 4 MHz genom att ersätta 5 MHz kristallen med en 8 MHz kristall, typ HC18 serie-resonans.
Internal memory:	-Three 28-pin sockets for EPROM, of which one can be used for EPROM or ByteWyde RAM. Max 24 K EPROM or 16 K EPROM and 2K/8K RAM. Use static ByteWyde RAM. Ex. MK4802 2K RAM. Use 2732 or 2764 EPROM. Note! the access time requirements with 4 MHz clock.(NOTE! On 1004-00, the chip in pos. 7B must be replaced to use 8K RAM).
Internt minne:	-Tre 28-stifts socklar för EPROM, av vilka en kan användas för EPROM eller ByteWyde RAM. Max 24 Kbytes EPROM eller 16K EPROM och 2/8K RAM. Använd statiskt ByteWyde RAM, ex. MK4802 2K. Använd 2732 eller 2764 EPROM. Obs! Access tiderna måste beaktas med 4 MHz. (Obs! På 1004-00 måste kretsen i pos. 7B bytas för att använda 8K RAM).
External memory:	-Expansion of the memory can be done up to 64 Kbytes of RAM/EPROM with a standard single board computer bus. Only static memories can be used. External memory in the same address range as the internal memory can not be accessed. With 4 MHz system clock, the access time limitations must be considered.
Externt minne:	-Minnet kan expanderas till max 64 Kbytes RAM/EPROM i en standard enkortsdator-buss. Endast statiska minnen kan användas. Externt minne på samma adresser som det interna minnet bör ej användas, men stör ej. Med 4 MHz systemklocka måste accesstids-begränsningarna beaktas.

Internal I/O:
Internt I/O

-Z80A-CTC. 4 counters with trigg-inputs and 3 Zero-count/Timeout outputs(Driving low 2 mA).

Z80A-SIO/2. 2 channels for serial data communication. TTL compatible levels (Driving low 2 mA output). Asynchronous as well as synchronous protocols, byte or bit oriented.

See the Z80A-SIO/2 product specification for details.

Code input. "1-of-8" input, directly connected to the internal data bus. Special strobe (INP 8) for activating one of the eight data bits or an external buffer circuit.

-Z80A-CTC. 4 räknare med trigg-ingångar och 3 Noll/Timeout utsignaler(Låg nivå sänker 2 mA)

Z80A-SIO/2. 2 kanaler för seriell data-kommunikation. TTL kompatibla nivåer. (Låg utgång sänker 2 mA). Asynkrona såväl som synkrona protokoll, byte eller bitorienterade.

Se Z80A-SIO/2 produktspecifikation för detaljer.

Kod-ingång. "1-av-8" ingång, direkt kopplat till den interna databussen. Speciell strobe (INP 8) för att aktivera en av åtta databitar eller en extern bufferkrets.

I/O-buffering:
I/O-buffer:

-The internal I/O can be buffered with the SIO-booster 5068 to achieve full V24 (RS232C) levels. The 5068 does not support the code-input however.

-Interna I/O kan buffras med SIO-boostern 5068 för att erhålla V24 (RS232C) signal-nivåer.5068 har inga kretsar för kod-ingången.

I/O-expansion:

-Max 64 I/O cards from the DataBoard series. Note the access time requirements with 4 MHz.

-Max 64 I/O-kort från DataBoard serie. Obs accesstiderna om 4 MHz klocka används.

Interrupts:

-2 levels. Active low.
NMI* Non-Maskable-Interrupt.
INT* Common level for I/O.

-2 nivåer. Aktivt låg insignal.
NMI* Non-Maskable-Interrupt (Ej avstängbart).
INT* Gemensam nivå för all I/O.

Watchdog:	-Time-out with CPU reset, unless pushed(INP6) by the program with 1 sec intervals or less. Can be disconnected by a jumper, or by connecting pin 3B to 0V in bus connector. -Time-out med CPU-reset, om inte återställning (INP6)sker med 1 sek. intervall eller mindre. Kan stängas av med bygling eller jordning i buskontakten.
Real-Time-Clock: Realtidsklocka:	-Provided through the on-board counter/timer chip (CTC), using the system clock or an external clock. -Erhålles med räknar/tidtagar-kretsen på kortet (CTC), som kan använda systemklockan eller en extern klocka.
Power: Kraftmatning:	+5 V +-5% 0.9 A (Excluding Memories) +5 V +-5% 0.9 A (Utan minneskretsar)
Connectors: Kontakter:	-B 64-pin Euroconnector plugs DIN 41612 on both bus-side and I/O-side. The I/O connector is keyed to prevent insertion in the bus. -B 64-stifts Europakontakt (plugg) DIN 41612 på både buss- och I/O-sidan. I/O-kontakten är nycklad för att förhindra att den stoppas in på bussen.
Environment: Omgivning:	Operating 0 - 55 degrees Celsius. 0 - 55 grader Celsius.

INTERNAL MEMORY

Three 28-pin memory sockets are available, two for EPROM only and one for RAM or EPROM according to the BYTEWYDE concept.

The 2732/2764 EPROM can be used in all three sockets. Note! The shorter access time in the ByteWyde socket. As RAM, use static ByteWyde RAM. Ex. MK4802 2K RAM.

The jumpers S1 --- S7 select the memory type and size. The EPROM socket at 4C is always at address 0000H followed by the EPROM socket at 5C in the address range. RAM in the socket 7C is always at high address below OFFFFH, while EPROM in the same socket is directly following the other EPROM sockets.

INTERNA MINNET

Tre 28-stifts minnessocklar finns på kortet, två för enbart EPROM och en för RAM eller EPROM av BYTEWYDE typ.

2732/2764 EPROM kan användas i alla tre socklarna. Obs. att accesstiden som krävs är kortare i ByteWyde-sockeln. Som RAM används statiskt Byte-Wyde RAM. Ex. MK4802 2K RAM.

Byglingarna S1 --- S7 väljer minnestyp och storlek. EPROM-kretsen i pos. 4C ligger alltid på address 0000H följd av EPROM-kretsen i pos. 5C i adressområdet. RAM i sockel 7C ligger alltid på höga adresser under OFFFFH, medan EPROM i samma sockel läggs direkt efter de andra EPROM kapslarna.

```
    OFFFFH                          OFFFFH
         I  RAM     7C   2K/8K           I
         I                               I   External
         I  External                     I
         I                               I   EPROM    7C   4/8K
         I                               I
         I  EPROM    5C  4/8K            I   EPROM    5C   4/8K
         I                               I
         I  EPROM    4C  4/8K            I   EPROM    4C   4/8K
     0000H                           0000H
```

NOTE! THE POSITION OF THE 24-PIN MEMORY CHIPS IN THE 28-PIN SOCKET!
See the component layout! The ground pin (no.12) shall, as the entire chip be adjusted towards the edge of the card.

The access time requirements on the internal memory chips are, depending on the clock:

OBS! POSITIONEN FÖR 24-STIFTS-MINNESKRETSEN I 28-STIFTS-SOCKELN!
Se komponentplaceringen. Jordstiftet (stift 12) skall, som hela kretsen sättas närmast kanten på kortet.

Accesstidskraven på de interna minneskretsarna är, beroende på systemklockan:

System clock		2.5MHz	4MHz	
In BYTEWYDE socket:	Tacc	655	355	nsec
	Tce	595	295	nsec
	Toe	448	223	nsec
In pure EPROM socket:	Tacc	655	355	nsec
	Tce	655	355	nsec
	Toe	423	198	nsec

EXTERNAL MEMORY

With 2.5 MHz system clock, standard DataBoard 4680 static memory boards with RAM and EPROM can be used,incl. the 2003 16K RAM.

With 4 MHz system clock, the access time limitations must be considered. Ex. The memory cards 2009/2019/3009 requires 200 ns memory circuits. Note that Z80 allows slower memories (longer access time) when pure data is accessed than for program code(Fetch).
The memory control signals on the DataBoard bus are:
MEMW* for memory write
MEMFL* for memory read
16 bits address
Wait-state generating for slow memories are not supported.

The external MEMFL* strobe is only generated for addresses outside the internal memory range.

Access time requirements.

Note. Given data holds for 1004 rev. D or later.

The total external access time depends on the bus delays, the delays on the selected memory cards and the delays in the memory chips.

The memory access time requirements depends on the system clock.

The access time requirements are as below on the bus connector. Longer access time is allowed at access of data than for program code(Fetch).

EXTERNT MINNE

Med 2.5 MHz systemklocka kan standard DataBoard 4680 minneskort användas med RAM och EPROM, inkl. 2003 16K RAM.

Med 4 MHz systemklocka måste accesstidskraven beaktas. T.ex. måste minneskorten 2009/2019/3009 använda 200 ns minneskretsar. Notera att Z80 har lägre krav (längre accesstid) för access av enbart data än för access av program kod(Fetch).
Minneskontrollsignalerna på DataBoard bussen är:
MEMW* skriva till minnet
MEMFL* läsa från minnet
16 bitars adress
"Wait-states" för långsamma minneskretsar genereras ej.

Den externa MEMFL* stroben genereras enbart för adresser utanför det interna minnesområdet.

Accesstidskrav.

OBS. Angivna data gäller för 1004 rev. D eller senare.

Den totala externa accesstiden bestäms av buss-fördröjningar, fördröjningar på de valda minneskorten samt fördröjningar i minneskretsarna.

Minnes-accesstids kraven beror på systemklockan.

Accesstidskraven är enligt nedan på busskontakten. Längre access tid tillåts vid access av data än för program kod(Fetch).

External memory access time.	Extern minnesaccesstid.	
System clock:	2.5 MHz	4 MHz
From ADDRESS to DATA:	625 nsec	325 nsec
From MEMFL* to DATA (Fetch):	371 nsec	146 nsec
From MEMFL* to DATA (Read data):	556 nsec	256 nsec

For external I/O cards, the access time requirements on the 1004 bus connector are:	För externa I/O kort gäller nedanstående accesstidskrav på 1004 busskontakt:	
System clock:	2.5 MHz	4 MHz
From INP-strobe to DATA:	821 nsec	446 nsec

DATABOARD BUS EXPANSION

Standard and flexible expansion is provided, using the 8-slot or the 20-slot backplanes. A user wired backplane according to the system manual can also be used.

The 1004 may use the double board computer backplane for the 1043/1044(Not 1045/46) if the CPU-slot is equipped with a 6061 jumper board and the 1004 is inserted in the control card slot. On backplanes with a MAC-slot, the 6054 jumper board must be installed in the MAC slot. See the system manual for details. Note! 1004 can not use the 4 MHz double board backplane unless inserted in an I/O-slot.

The 8-slot backplane is for single board computers and ABC80/ABC800 only.

NOTE! that the SENSE* input strobe and the general OUTSTB* output strobe are available on the bus connector. If an old version of the double board computer bus is used, the pin 25A (SENSE*) on the bus must be disconnected from the ADRFL* (CPU-slot pin 25A). Old versions of the single-board computer backplane does not have the SENSE* strobe wired on the I/O-bus.

DATABOARD BUSEXPANSION

En standardiserad och flexibel expansion erhålles med 8-positions eller 20-positions bakplan. Ett bakplan virat av användaren enligt system-manualen kan även användas.

1004 kan använda tvåkorts-datorns bakplan för 1043/1044 (Ej 1045/46), om 6061 byglingsplugg installeras i CPU-positionen och 1004 sättes i kontroll-kortets position. I bakplan med MAC-position måste 6054 byglingsplugg sättas i MAC-positionen.
Se systemmanualen för detaljer. Obs! 1004 kan ej använda 4 MHz tvåkortsdatorbakplan, om den ej sättes i en I/O-position.

8-positions bakplanet är enbart för enkortsdatorer samt ABC80/ABC800.

OBS! att SENSE* in-stoben samt den generella OUTSTB* ut-stoben finns på buss-kontakten. Om en äldre variant av tvåkortsdatorbussen används, måste stift 25A (SENSE*) på bussen kopplas bort från ADRFL* (CPU-pos. stift 25A).
Äldre versioner av enkorts-datorbussen har ej SENSE*-stroben virad på I/O-bussen.

INTERRUPT

Two interrupt levels are available.

- NMI*
The Non-Maskable-Interrupt is directly connected from the bus connector. An active low signal generates a call to address 66H. This can not be disabled by software and is used for actions needing fast response, like the "Power-Down" routine. Return from an NMI service routine shall use the RTN instruction.

- INT*
The interrupt is permanently connected from the internal I/O circuits CTC and SIO. They are connected in a priority chain "daisy chain", to which the external interrupt input can be connected by the jumpers S9 and S10.

The priority order is from the highest to the lowest:

INTERRUPT

Två interrupt-nivåer finns.

- NMI*
Det ej maskbara interruptet är direkt anslutet till buss-kontakten. En aktivt låg signal genererar ett anrop till adress 66H. Detta kan ej stängas av med programvara och används för rutiner som kräver snabba åtgärder, som t. ex. kraftavbrottsrutiner.
Vid return från en NMI rutin skall RTN instruktionen användas.

- INT*
Interrupt är alltid anslutet från de interna I/O-kretsarna CTC och SIO. De är anslutna i en prioritetskedja "daisy-chain", till vilken den externa interruptingången kan anslutas med byglingarna S9 och S10.

Prioritetsordningen är från den högsta till den lägsta:

CTC0, CTC1, CTC2, CTC3, SIO-A, SIO-B, External

To reset the priority chain after servicing an internal interrupt, the RTI instruction should be used.

The Z80 interrupt is mode controlled by the program. At power-up, mode 0 is automatically set and interrupts are disabled.

Mode 1:
Interrupt generates a call to the address 38H.

Mode 2 and Mode 0:
The interrupting device generates a vector on the data bus. For the CTC and SIO, the user program has stored the vector in a register. For external interrupts the vector 0FFH is always generated.

The CTC can not be used with mode 0, as the vector bit 0=0.

För att återställa prioritets-kedjan efter att ett internt interrupt åtgärdats, ska RTI instruktionen användas.

Z80 interrupt struktur är mod-styrt av programmet. Vid spänningstillslag sättes automatiskt mod 0 och interrupt avkänningen är avstängd.

Mod 1:
Interrupt genererar ett sub-rutinanrop till adress 38H.

Mod 2 och Mod 0:
Den enhet som gett interrupt, genererar en vektor på data-bussen. För CTC och SIO har användarprogrammet lagrat vektorn i ett register. För externa interrupt genereras alltid vektorn 0FFH.

CTC kan ej användas med mod 0, eftersom vektorns bit 0 = 0.

Mode 0: (Not when CTC is used)
The Z80 interprets the vector as a one-byte instruction. Normally a RST-instruction is used to generate a call to one of the memory addresses below. The corresponding vectors are also given:

Mod 0: (Inte då CTC används)
Z80 tolkar vektorn som en en-bytes-instruktion. Normalt används en RST-instuktion för att generera ett anrop till en av minnesadresserna nedan. De motsvarande vektorerna ges även:

Address:	00H	08H	10H	18H	20H	28H	30H	38H	
Vector:	0C7H	0CFH	0D7H	0DFH	0E7H	0EFH	0F7H	0FFH	(0FFH=external)

Mode 2:
The vector, together with the I-register in the CPU, is a 16-bit pointer to an address-table, containing the entry addresses to interrupt service routines. Bit 0 in the vector is set to 0 by Z80 in mode 2.

The I-register and the address are generated by the user.

Mod 2:
Vektorn används tillsammans med I-registret i CPU som en 16-bits pekare till en adress-tabell med pekare till sub-rutiner för interrupthantering. Bit 0 i vektorn sätts till 0 av Z80 i mod 2.

I-registret och adresstabellen skrivs av användaren.

Vector (low-byte) + I-reg (high byte) ----> Table with pointer.
Pointer ----> Entry address to subroutine.

WATCHDOG

Program control by hardware is achieved by the "watch-dog" circuit, which resets the CPU unless regularly pushed by an INP 6 instruction in the user program. The watch-dog shall be pushed with a period of less than 1 second, set by the components R5 and C4.

The jumper S8 is closed to permanently disconnect the watch-dog, which also can be achieved by holding KILLDOG* low (pin 3B in the bus connector).

No I/O-RST* signal is generated by the watch-dog.

WATCHDOG

Övervakning av programmet med hårdvara erhålles med "watch-dog" kretsen, som ger reset av CPU om den inte regelbundet "stötes till" med en INP 6 instruktion i programmet. Påstötning ska ske med en period mindre än 1 sekund, bestämt av komponenterna R5 och C4.

Bygling S8 slutes för att permanent stänga av watch-dog, vilket även kan göras genom att hålla KILLDOG* låg (pin 3B i buskontakten).

Ingen I/O-RST* signal genereras av watch-dog.

RESTART

At power-up or hardware reset by the RESIN* signal, the CPU, the CTC and the SIO are reset and an I/O-RST* signal resets all I/O-cards on the DataBoard I/O-bus. The I/O-RST* can also be generated by the INP 7 instruction.

UPP- OCH ÅTERSTART

Vid spänningspåslag och hård-varureset med RESIN* signalen, ges reset till CPU,CTC och SIO och en I/O-RST* signal går ut till alla I/O-kort på Data-Board I/O-buss. I/O-RST* kan även genereras av INP 7 - instruktionen.

STANDARD DATABOARD I/O-CONTROL

For the DataBoard 4680 I/O bus, the standard I/O strobes are generated. See the system manual for details.

OUT*	OUT 0	DATA
CS*	OUT 1	Card Select
C1*	OUT 2	Out strobe
C2*	OUT 3	Out strobe
C3*	OUT 4	Out strobe
C4*	OUT 5	Out strobe

OUTSTB* General OUT-strobe, generated at all OUT-XXX

Note that the SENSE* strobe (INP 2) is not used in older DataBoard systems. It corresponds to OP5* on pin 25A.

With 2.5 MHz system clock, all DataBoard I/O-cards are available, but with 4 MHz, the access time requirements should be considered for complex interfaces.See the table in the section about external memory.

STANDARD DATABOARD I/O-STYRNING

Standardiserade I/O-strobar genereras för DataBoard 4680 I/O-buss. Se system manualen för detaljer.

INP*	INP 0	DATA
STAT*	INP 1	Status input
SENSE*	INP 2	New in-strobe
PUSHDOG*	INP 6	PUSH watchdog
I/O-RST*	INP 7	Reset I/O bus

Generell OUT-strob, genereras vid alla OUT-XXX

Obs! att SENSE* stroben (INP2) inte användes i äldre DataBoard system. Den motsvarar OP5* och finns på stift 25A.

Med 2.5 MHz systemklocka är alla DataBoard I/O-kort användbara, men med 4 MHz bör accesstidskraven beaktas för komplexa interface. Se tabellen i kapitlet om extern minne.

CONTROL OF INTERNAL I/O

For the internal I/O, i.e. the CTC, SIO and the "1 of 8" code the following I/O port are used.

CTC0	OUT 30H	INP 30H	
CTC1	OUT 31H	INP 31H	
CTC2	OUT 32H	INP 32H	
CTC3	OUT 33H	INP 33H	
SIO-A	OUT 38H	INP 38H	Data I/O
SIO-A	OUT 39H	INP 39H	Control
SIO-B	OUT 3AH	INP 3AH	Data I/O
SIO-B	OUT 3BH	INP 3BH	Control
Code		INP 8	1 of 8 code input. (Port 8,9,..0FH gives the same)

The external DataBoard strobes are independant of the internal I/O control strobes. Thus the card selection on the bus is not effected by internal I/O.

STYRNING AV INTERN I/O

För intern I/O, dvs. CTC, SIO och "1 av 8" avkodaren gäller följande I/O-portar.

De externa DataBoard strobarna är oberoende av de interna I/O styrningen. Till exempel påverkas inte kortvalet på bussen av intern I/O.

CTC CONTROL + EXAMPLE

NOTE! Below is only a short description. The user needs the ZILOG CTC product specification in addition to this datasheet!

Programming a CTC counter/timer is done by writing two or three control bytes to the CTC.

Each CTC can be used as a down-counter of incoming signals. Alternatively it can be used as a timer, counting scaled system clock periods. If selected, the timer can be trigged by an external signal.

CTC STYRNING + EXEMPEL

OBS! Nedan är endast en kort beskrivning. Användaren behöver även ZILOG´s CTC produktbeskrivning.

Att programmera en CTC räknar/tidtagarkrets görs genom att skriva två eller tre kontrolltecken till CTC.

Varje CTC kan användas som en nedräknare av inkommande pulser. Alternativt kan den användas som en tidtagarkrets,som räknar nedskalade systemklockspulser. Tidtagaren kan triggas av en extern signal, om detta valts.

Channel control byte. Bit 0 = 1.

	D7	D6	D5	D4	D3	D2	D1	D0
1/ 0	Int. enable	Count/ Timer	Pre- scale 256/16	Rising/ Falling edge	Trigg	Time const. follows	Reset	1

D5 and D3 are only used in timer mode.

D5 och D3 används enbart i tidtagarmod.

Time constant byte.: 1 - 255, 0=256.

This constant is the start value in the down-conter.

Denna konstant är startvärdet i nedräknarregistret.

Interrupt vector. Bit 0 = 0. Written to CTC0-port.

D7	D6	D5	D4	D3	D2	D1	D0
<--User supplied vector------>					X	X	0

The interrupt vector is always written to the CTC-0 port. At interrupt, the CTC inserts the CTC number (0-3) in the bits D1,D2.

As bit 0 in the vector is 0, the CTC can not give the one-byte RST n instructions used in the Z80 interrupt mode 0.

Interrupt vektorn skrivs alltid till CTC-0 porten. Vid interrupt sätter CTC bitarna D1,D2 till CTC-numret (0-3).

Då bit 0 i vektorn är 0, kan CTC inte ge någon en-byte RST n instruktion i Z80 interruptmod 0.

The user can read the current contents of the down-counter register by an INP instruction.

Användarprogrammet kan läsa värdet i ned-räknar-registret med en INP instruktion.

Example of CTC programming

Initiate the CTC-1 as a timer, starting to count down at the rising edge of a pulse on the CTC-1 Trigg-input. The timer shall count the system clock pulses, scaled down with the factor 256. The down-counter start value in 15, making the counter count 256*15 clock pulses before initiating an interrupt and outputing a signal (time-out) on the CTC-1 output. The address to the interrupt routine is stored in the memory position 0F2H,0F3H for the CTC-1. The interrupt vector written to CTC-0 is then 0F0H, while the I-register in the CPU must be set to 00 and the interrupt mode 2 selected.

Write to CTC0:port 30H
Note! always CTC0!
Interrupt vector: 0F0H

Set CPU I-reg. to 0.
LI A,00
LR I,A
Select interrupt mode 2.
IM 2

Write to CTC1: port 31H.
Channel control byte 0BDH
(Interrupt ON,Timer,Prescaler 256,Rising edge, Trigger on,Start value follows)

Time constant byte: 15

Exempel på CTC programmering.

Initiera CTC-1 som tidtagare, som ska starta nedräkningen vid en uppåtgående signal på CTC-1 Trigg-ingång. Kretsen ska räkna systemklockan, nedskalad med faktorn 256. Startvärdet för nedräknaren är 15, vilket ger en nedräkning med 256*15 klockpulser innan interrupt genereras och en signal (time-out) sändes på CTC-1 utgången. Adressen till interruptrutinen ligger i minnesadressen 0F2H-0F3H för CTC-1. Interrupt vektorn, som ska skrivas till CTC-0, blir då 0F0H, medan I-registret i Z80-CPU sättes till 00 och interrupt mod 2 väljs.

Skriv till CTC0: port 30H
OBS! alltid CTC0
Interrupt vektor: 0F0H

Sätt CPU I-reg. till 0.
LI A,00
LR I,A
Välj interrupt mode 2.
IM 2

Skriv till CTC1: port 31H
Kanalkontrollbyte: 0BDH
(Interrupt ON, Tidtagare, Prescaler 256,Positiv flank, Trigger ON,Startvärde följer)

Räknarens startvärde: 15

SIO CONTROL + EXAMPLE

NOTE! Below is only a short description. The user needs the ZILOG SIO/2 product specification in addition to this datasheet.

The SIO/2 circuit is a programmable, dual channel device, which provides formatting of data for serial data communication.

It is capable of handling asynchronous, synchronous and bit oriented synchronous protocols, as IBM BiSync, HDLC, SDLC and virtually any other serial protocol.
It can generate CRC codes in any synchronous mode and can be programmed by the CPU for any traditional asynchronous formats.

The transmit and recieve clocks are taken from the I/O-connector, where jumpers can select on-board clock sources.

The transmit and the recieve clocks are independantly selected for both channels as the SIO/2 version is used.

Only channel A has a SYNC input.

The clocks can be from:

- External sources
- System clock and the CTC, which in one or more steps generates the clockpulses. The CTC pulses are not symmetric.
- System clock, which with half the frequency is available on the I/O-connector.

The clock source can be feed through an on-board flip-flop to achieve a symmetric clock.

See the block-diagram for the pin numbering.

SIO-STYRNING + EXEMPEL

OBS! Nedan är endast en kort beskrivning. Användaren behöver även ZILOG´s SIO/2 produktbeskrivning.

SIO/2 kretsen är en programmerbar tvåkanalsenhet, som utför formattering av data för seriell datakommunikation.

Den kan hantera asynkrona, synkrona och bitorienterade synkrona protokoll, såsom IBM BiSync, HDLC, SDLC och praktiskt taget vilket annat seriellt protokoll som helst.
Den kan generera CRC-kod i alla synkrona protokoll och kan programmeras av CPU till alla normala asynkrona format.

Sändnings och mottagningsklockorna tas från I/O-kontakten, där byglingar kan välja klocksignaler från kortet.

Sändnings och mottagningsklockorna är oberoende av varandra i båda kanalerna då SIO/2 versionen används.

Endast kanal A har SYNC ingång.

Klockorna kan tas från:

- Externa klockor.
- Systemklockan och CTC, som i ett eller flera steg genererar klockpulserna. CTC pulserna är inte symmetriska.
- Systemklockan, som med halva frekvensen är tillgänglig i I/O kontakten.

Klockpulserna kan byglas att passera en flip-flop på kortet för att få en symmetrisk klocka.

Se blockdiagrammet för stiftsnumreringen.

Programming the SIO/2 is done by writing control bytes to one of 8 "write registers" for each channel.(WR2 is always written to channel B).

Progammering av SIO/2 görs genom att skriva kontroll-tecken till ett av 8 "skriv-register" för vardera kanalen. (WR2 skrivs enbart till kanal B).

Status information is read from one of 3 "read-registers" for each channel(RR2 is only read from channel B).

Status information läses från ett av 3 "läsregister" för vardera kanalen (RR2 läses enbart från kanal B).

To read from or write to a register, other than RR0 or WR0, a register pointer must be written to WR0, bits 0-2. While writing the pointer, the bits 3-7 will be stored in WR0, unless they are ZER0.

För att läsa från eller skriva till ett register, förutom RR0 och WR0, måste en register-pekare skrivas till WR0,bit0-2. Då pekaren skrivs, kommer bitarna 3-7 att skrivas i WR0, om de inte är NOLL.

The register pointer must be written each time any register, except RR0 or WR0, shall be accessed, as the pointer is automatically cleared after each access, as well as at a reset.

Registerpekaren måste skrivas varje gång ett register, utom RR0 eller WR0 skall accessas, eftersom pekaren automatiskt nollas efter varje access, liksom vid reset.

Below is a short description of the SIO control registers.

Nedan följer en kort beskriv-ning av SIO kontroll-registren.

For more details, see the Z80A-SIO/2 product specification.

För mer detaljer, se Z80A-SIO/2 produkt specifikation.

SIO/2 CONTROL REGISTERS

-Write registers, controlling the function. One set per channel except WR2, which is always written to channel B.

WR0: Direct commands and a register pointer,selecting register to access with the next command.

WR1: Transmit/Receive data and interrupt mode definition.

WR2: Interrupt vector. Only one, common for the card, stored as a channel B register.

WR3: Receive parameters and control.

WR4: Transmit/Receive modes and parametrar.

WR5: Transmit parameters and control.

WR6: Sync. character or SDLC address field.

WR7: Sync. character or SDLC flag.

-Read registers, with status information. One set per channel except RR2, which is only read from channel B.

RR0: Transmit/Receive buffer status.

RR1: Special receive status.

RR2: Modified Interrupt Vector, always read from channel B.

SIO/2 KONTROLLREGISTER

-Skrivregister, som styr funktionen. En uppsättning per kanal utom WR2, som alltid skrivs till kanal B.

WR0: Direktkommandon samt registerpekare, som pekar på det register som skall användas i nästa kommando.

WR1: Sänd/Mottagning data och interrupt mod definition.

WR2: Interrupt vektor. Endast en gemensam för kortet, lagras som ett kanal B register.

WR3: Parametrar och styrning av mottagning.

WR4: Sänd/Mottagnings mod och parametrar.

WR5: Parametrar och styrning av sändning.

WR6: Sync. tecken eller SDLC adress fält.

WR7: Sync. tecken eller SDLC flagga.

-Läsregisters med status information. En uppsättning per kanal utom för RR2, som endast läses från kanal B.

RR0: Sänd/Mottagnings buffer status.

RR1: Särskild mottagnings-status.

RR2: Modifierad interrupt vektor. Läses alltid från kanal B.

```
REGISTER BIT ASSIGNMENT                REGISTER BIT FUNKTIONER
 ...................................................................
READ REGISTER 0                        LÄSREGISTER 0
---------------                        -------------
7  6  5  4  3  2  1  0  Bit
                     x--Rx character available
                  x-----INT pending (Read from Ch.A only)
               x--------Tx buffer empty
            x-----------DCD
         x--------------SYNC/HUNT
      x-----------------CTS
   x--------------------Sending CRC/SYNCS
x-----------------------Break/Abort
 ...................................................................
READ REGISTER 1                        LÄSREGISTER 1
---------------                        -------------
7  6  5  4  3  2  1  0  Bit
                     x--All sent

                        I-field bits     I-field bits in
                        in previous      second previous
                        byte             byte
            1  0  0         0                3
            0  1  0         0                4
            1  1  0         0                5      Residue
            0  0  1         0                6      data for
            1  0  1         0                7      8 Rx bits/
            0  1  1         0                8      character
            1  1  1         1                8      programmed.
            0  0  0         2                8

         x--------------Parity error
      x-----------------Rx overrun error     Special Rx
   x--------------------CRC/Framing error    condition
x-----------------------End-of-Frame(SDLC)   interrupts
 ...................................................................
READ REGISTER 2                        LÄSREGISTER 2
---------------                        -------------
(Always read from Ch.B)                (Läses alltid från kanal B)

7  6  5  4  3  2  1  0  Bit
                     x--V0
                  x-----V1
               x--------V2    Interrupt vector
            x-----------V3    Modified if STATUS-AFFECTS-VECTOR
         x--------------V4    is set!
      x-----------------V5
   x--------------------V6
x-----------------------V7
```

```
REGISTER BIT ASIGNMENT                REGISTER BIT FUNKTIONER
 ...................................................................
WRITE REGISTER 0                      SKRIVREGISTER 0
----------------                      ---------------
7  6  5  4  3  2  1  0  Bit
               0  0  0  Register 0 pointer
               0  0  1            1
               0  1  0            2
               0  1  1            3
               1  0  0            4
               1  0  1            5
               1  1  0            6
               1  1  1            7

      0  0  0-----------Null code (no action)
      0  0  1-----------Send abort (SDLC)
      0  1  0-----------Reset Ext. status interrupts
      0  1  1-----------Channel reset (NOTE! Takes 4 timestates)
      1  0  0-----------Reset RxINT on first character
      1  0  1-----------Reset TxINT pending
      1  1  0-----------Error reset
      1  1  1-----------Return from interrupt (CH.A only)
0  0--------------------Null code
0  1--------------------Reset Rx CRC checker
1  0--------------------Reset Tx CRC generator
1  1--------------------Reset CRC/SYNCS sent/sending latch

 ...................................................................
WRITE REGISTER 1                      SKRIVREGISTER 1
----------------                      ---------------
7  6  5  4  3  2  1  0  Bit
                     x--Ext. INT enable
                  x-----Tx INT enable
               x--------STATUS-AFFECTS-VECTOR (Ch.B only!)
         0  0-----------RxINT disable
         0  1-----------RxINT on first char. only or error.
         1  0-----------INT on all Rx characters(Parity affects
                        vector.)
         1  1-----------INT on all Rx characters(Parity does
                        not affect vector.)
      x-----------------Wait/Ready on R/T
   x--------------------Wait FN/Ready FN
x-----------------------Wait/Ready enable

 ...................................................................
WRITE REGISTER 2                      SKRIVREGISTER 2
----------------                      ---------------
(Only written to channel B)           (Skrivs bara till kanal B)

7  6  5  4  3  2  1  0  Bit
                     x--V0
                  x-----V1
               x--------V2      Interrupt vector
            x-----------V3      V1,V2,V3 modified at interrupt
         x--------------V4      only IF the "Status-Affects-
      x-----------------V5      Vector" bit has been set in WR1.
   x--------------------V6      In Z80 int. mode 2, V0 must be 0.
x-----------------------V7
```

```
WRITE REGISTER 3                    SKRIVREGISTER 3
----------------                    ---------------
7  6  5  4  3  2  1  0  Bit
                     x--Rx enable
                  x-----Sync character load inhibit
               x--------Address search mode (SDLC)
            x-----------Rx CRC enable
         x--------------Enter hunt mode
      x-----------------Auto enables.
0  0--------------------Rx 5 bits/character
0  1--------------------Rx 7 bits/character
1  0--------------------Rx 6 bits/character
1  1--------------------Rx 8 bits/character
 ..............................................................
WRITE REGISTER 4                    SKRIVREGISTER 4
----------------                    ---------------
7  6  5  4  3  2  1  0  Bit
                     x--Parity enable
                  x-----Parity EVEN(1)/ODD(0)
            0  0--------Sync modes enable
            0  1--------1 stop bit/character
            1  0--------1 1/2 stop bit/character
            1  1--------2 stop bits/character
      0  0--------------8 bit sync character
      0  1--------------16 bit sync character
      1  0--------------SDLC mode (0111 1110 Sync flag)
      1  1--------------External sync mode
0  0--------------------x1 clock mode
0  1--------------------x16 clock mode
1  0--------------------x32 clock mode
1  1--------------------x64 clock mode
 ..............................................................
WRITE REGISTER 5                    SKRIVREGISTER 5
----------------                    ---------------
7  6  5  4  3  2  1  0  Bit
                     x--Tx CRC enable
                  x-----RTS
               x--------SDLC/CRC-16
            x-----------Tx enable
          x-------------Send Break.
   0  0-----------------Tx 5 bits (or less)/character
   0  1-----------------Tx 7 bits/character
   1  0-----------------Tx 6 bits/character
   1  1-----------------Tx 8 bits/character
x-----------------------DTR
==============================================================
WRITE REGISTER 6                    SKRIVREGISTER 6
-----------------------------       -----------------------------
7  6  5  4  3  2  1  0  Bit
                     x--Sync bit 0
                  x-----Sync bit 1       Also SDLC address
         ..... etc..                     field.
x-----------------------Sync bit 7
 ..............................................................
WRITE REGISTER 7                    SKRIVREGISTER 7
-----------------------------       -----------------------------
7  6  5  4  3  2  1  0  Bit
                     x--Sync bit 8
                  x-----Sync bit 9      For SDLC it must be
     ....etc.........                   0111 1110 for flag
x-----------------------Sync bit 15    recognition.
```

HANDLING SIO-INTERRUPTS

-In the SIO/2 an Interrupt Vector can be stored in the register WR2, with channel B selected. This vector holds for all functions on the card, but the SIO/2 can be programmed to modify the interrupt vector when sending it to the Z80. With this modification different interrupt service routines can be called, depending on the type of interrupt.

-The STATUS-AFFECTS-VECTOR bit in channel B,WR1,BIT 1 indicates if the vector shall be modified by the SIO depending on the cause of the interrupt.

-Bits 1-3 in the modified interrupt vector indicate:

Bit 3 :	0xxx	--	Channel B
	1xxx	--	Channel A
Bit 1,2 :	x00x	--	Transmit buffer empty interrupt
	x01x	--	External/Status change interrupt
	x10x	--	Receive character available
	x11x	--	Special receive condition (Parity,Rx-overrun or CRC/Framing error or End-of-Frame(SDLC))

-In the Z80 interrupt mode 2 the CPU always sets the bit 0 in the vector to 0.

-When an interrupt has been detected, the channel A register RR0 bit 1 can be tested. This bit indicates if the SIO (ch.A or ch.B) has sent an interrupt or not.

HANTERING AV SIO-INTERRUPT

-I SIO/2 kan en Interruptvektor lagras i registret WR2, med kanal B vald. Denna vektor gäller alla funktioner på kortet, men SIO/2 kan programmeras att modifiera interrupt vektorn innan den sänds till Z80. Med denna modifikation kan olika interrupt rutiner anropas beroende på vilken typ av interrupt som skett.

-STATUS-AFFECTS-VECTOR biten i kanal B, WR1, bit 1 anger om vektorn ska modifieras av SIO beroende på orsaken till interruptet.

-Bitarna 1-3 i den modifierade interruptvektorn betyder:

-I Z80 interrupt mod 2 sätter alltid CPU bit 0 i vektorn till 0.

-När ett interrupt har detekterats kan kanal A register RR0 bit 1 testas. Denna bit indikerar om SIO (kanal A eller B) har sänt interrupt eller ej.

Example of SIO programming

See examples in appendix 1.
One of the exemples is a
a program in assembler,
using the CTC and SIO to simulate a normal UART in/out port.

The baudrate is 9600 Baud.
The program reads from port A
and echoes the character back.
One version is with interrupt
and another without interrupt.
The interrupt version also
echoes direct to a usual
9600 Baud UART interface.

In the 1004-H examples the
CTC-2 output shall be wired on
the I/O connector to the clock
inputs on the SIO channel A
(T*C and R*C).
I/O pin 8B to pins 25B and 27B.

Exempel på SIO programmering

Se exempel i appendix 1.
Ett av exemplen är ett
assemblerprogram, som
med CTC och SIO simulerar
en normal UART in/ut port.

Baudrate är 9600 Baud.
Programmet läser från port A
och ekar tillbaka tecknet.
En version med interrupt och
en utan interrupt finns med.
Interruptversionen ekar även
direkt till ett vanligt
9600 Baud UART interface.

I exemplen i 1004-H skall
CTC-2 utgången viras i
I/O-kontakten till klockingångarna på SIO kanal A
(T*C and R*C).
I/O stift 8B till 25B och 27B.

CODE INPUT CONTROL

The I/O-pins 9A,10A,....16A are directly connected to the internal databus.

An INP 8 instruction pulls the strobe (pin 18A) active low, while the CPU reads the databus.

By wiring ONE jumper between the strobe (18A) and ONE of 9A,...16A, the CPU reads a byte with one bit set to "0" and all other bits "1".
DO NOT connect the databus bits with each other.

If a general 8-bit value shall be read, a logical gate must be installed close to the I/O-connector (within 50 mm), activated by the strobe.

STYRNING AV KODINLÄSNING

I/O-stiften 9A,10A,....16A är direkt anslutna till den interna databussen.

En INP 8 instruktion drar stroben (stift 18A) aktivt låg, medan CPU läser databussen.

Genom att vira EN bygel mellan stroben (18A) och EN av 9A,....16A, läser CPU en byte med en bit satt till "0" och alla andra bitar "1".
BYGLA EJ samman databitar med varandra.

Om ett generellt 8-bitars värde skall läsas måste en logisk grind installeras nära I/O-kontakten(inom 50 mm), och aktiveras av stroben.

I/O-PIN NUMBERING
The '*' after a signal name indicates that the signal is active low.

I/O-STIFTSNUMRERING
Stjärnan '*' efter ett signalnamn indikerar att signalen är aktivt låg.

INT*	30A	In	External interrupt Parallel to bus-interrupt.	Extern interrupt Parallell med buss-interrupt.
INT	29A	Out	External interrupt	Extern interrupt.
INT	25A	I/O	See block diagram.	Se blockschemat.
CTC0-Trigg	5B	In	IN= Counter pulses or Timer start. pulse.	IN=Räknepulser eller start puls till timer.
CTC0-OUT	6B	Out		
CTC1-Trigg	6A	In		
CTC1-OUT	7B	Out	OUT=Zero count or Timeout. (Active high pulse)	UT=Nedräkning till noll eller Timeout puls (Aktivt hög)
CTC2-Trigg	7A	In		
CTC2-OUT	8B	Out		
CTC3-Trigg	8A	In		
SIO-A SYNC	19B	In	A-Sync	A-Sync
TxC	25B	In	A-Transmit clock	A-Sändningsklocka
RxC	27B	In	A-Receive clock	A-Mottagningsklocka
TxD	12B	Out	A-Data Out	A-Data utgång
RxD	13B	In	A-Data In	A-Data ingång
RTS	14B	Out	A-Req.To Send	A-Begär Sändning
CTS	15B	In	A-Clear to Send	A-Klart att sända
DTR	30B	Out	A-Dev.Term.Ready	A-Enheten klar
DCD	18B	In	A-Data Carrier Det.	A-Bärvåg klar
SIO-B TxC	21B	In	B-Transmit clock	B-Sändningsklocka
RxC	26B	In	B-Receive clock	B-Mottagningsklocka
TxD	24B	Out	B-Data Out	B-Data utgång
RxD	20B	In	B-Data In	B-Data ingång
RTS	29B	Out	B-Req.To Send	B-Begär sändning
CTS	23B	In	B-Clear to Send	B-Klart att sända
DTR	28B	Out	B-Dev.Term.Ready	B-Enheten klar
DCD	22B	In	B-Data Carrier Det.	B-Bärvåg klar
Code Strobe*	18A	Out	INP 8 strobe	INP 8 strobe
Code Data0	16A	In	Databus direct	Databuss direkt
Data1	15A		-"-	-"-
2	14A			
3	13A			
4	12A			
5	11A			
6	10A			
Data7	9A		-"-	-"-
Syst.clock/2	5A	Out	Syst.clock/2	Systemklockan/2
FlipFlop IN	3B	In	Clock in	Klocka in
Q-out	4A	Out	Clock/2 out	Klocka/2 ut
Q* out	3A	Out	Clock/2* out	Klocka/2* ut

Signal ground and power. Signaljord och spänningar.

+12V	32A/32B		
+ 5V	31A/31B		
0V	2A/2B	Signal ground	Signaljord
-12V	1A/1B		

JUMPERS

The jumpers S1,S2,S3,S4,S5, S6 and S7 select the memory type and size.

BYGLINGAR

Byglingarna S1,S2,S3,S4,S5, S6 och S7 väljer minnestyp och storlek.

Pure EPROM sockets	ByteWyde socket	S1	S2	S3	S4	S5	S6	S7
2 * 2732 (0-8K)	2K RAM (62-64K)	OFF	ON	ON	OFF	ON	ON	OFF
2 * 2732 (0-8K)	8K RAM (56-64K)	ON	ON	ON	OFF	ON	OFF	ON
2 * 2764 (0-16K)	2K RAM (62-64K)	ON	OFF	ON	OFF	ON	ON	OFF
2 * 2764 (0-16K)	8K RAM (56-64K)	ON	ON	OFF	OFF	ON	OFF	ON
No EPROM	2K RAM (62-64K)	OFF	OFF	ON	OFF	ON	ON	OFF
2 * 2732 (0-8K)	2732 (8-12K)	OFF	ON	OFF	OFF	ON	OFF	ON
2 * 2764 (0-16K)	2764 (16-24K)	ON	OFF	OFF	OFF	ON	OFF	ON
2 * 2764 (0-16K)	2764 (24-32K)	OFF	OFF	OFF	OFF	ON	OFF	ON

External RAM cards, like the 2009 8K RAM card, can be placed in parallel with the internal RAM. Then the internal RAM is used for the highest addresses.

Externa RAM-kort, såsom 2009 8K RAM, kan placeras parallellt med det interna RAM-et. Det interna RAM-et används då för de högsta adresserna.

The jumper S8 disconnects the watch-dog.

S8 ON Watch-dog disconnected.
S8 OFF Enable watch-dog.

Bygling S8 kopplar bort watch-dog.

S8 ON Watch-dog urkopplad.
S8 OFF Watch-dog inkopplad.

The jumpers S9 and S10 connects the external interrupts.

S9:ON S10:OFF disables the external interrupts.
S9:OFF S10:ON connects the external interrupts with lowest priority in the "daisy-chain".

For functional compatibility with the 1062 single board computer, S9:ON and S10:OFF is required.

Byglingarna S9 och S10 ansluter externa interrupt.

S9:ON S10:OFF kopplar bort externa interrupt.
S9:OFF S10:ON kopplar in externa interrupt med lägsta prioritet i kedjan.

Funktionskompatibilitet med enkortsdatorn 1062 kräver S9:ON och S10:OFF.

For the earlier 1004-00, the following table was used. The difference is the chip in position 7B.
NOTE!! NOT FOR 1004-10 !

För den tidigare 1004-00 används nedanstående tabell. Skillnaden är kretsen i position 7B.
OBS!! INTE FÖR 1004-10 !

Pure EPROM sockets	ByteWyde socket	S1	S2	S3	S4	S5	S6	S7
2 * 2716 (0-4K)	2K RAM (62-64K)	ON	ON	ON	ON	OFF	ON	OFF
2 * 2732 (0-8K)	2K RAM	OFF	ON	ON	OFF	ON	ON	OFF
2 * 2764 (0-16K)	2K RAM	ON	OFF	ON	OFF	ON	ON	OFF
No EPROM	2K RAM (62-64K)	OFF	OFF	ON	x	x	ON	OFF
2 * 2716 (0-4K)	2716 *)(4-6K)	ON	ON	OFF	ON	OFF	OFF	ON
2 * 2732 (0-8K)	2732 (8-16K)	OFF	ON	OFF	OFF	ON	OFF	ON
2 * 2764 (0-16K)	2764 (16-24K)	ON	OFF	OFF	OFF	ON	OFF	ON
No on-board memory		OFF	OFF	OFF	x	x	x	x

*) Not 4 MHz x = Don´t care = Betydelselös

PROGRAM EXAMPLES

1. CETEST

Assembler program. Example of the use of SIO in asynchronous mode.
Two examples are included, with or without interrupt.

2. SIO1

BASIC program (Prommable), how to init SIO and CTC. Reads lines from channel A and sends back the line through the same channel.

3. SIOINUT

Assembly routines for reading from channel A and writing to channel B.

4. SIO2

BASIC program (Prommable), as SIO1 above, but using the SIOINUT in PROM for reading and writing the channels A and B.

5. SIOUART

Assembly program for control of SIO and CTC as well as a UART card with interrupt.
Written for use from prommable BASIC.
Read-routine for channel A, write-routine for channel B and read routine for a UART card.

6. AVBROTT

BASIC program (Prommable)
Init CTC and SIO.
Uses the SIOUART routines.
Reads lines both from UART and SIO-channel A. Sends the lines to SIO-channel B.

PROGRAM EXEMPEL

1. CETEST

Assembler program. Exempel på användning av SIO i asynkron mod.
Två exempel finns, med eller utan interrupt.

2. SIO1

BASIC program (Prommerbart), hur man initierar SIO och CTC. Läser rader från kanal A och sänder tillbaka raden till samma kanal.

3. SIOINUT

Assebler rutiner för att läsa från kanal A och skriva till kanal B.

4. SIO2

BASIC program (Prommerbart), som SIO1 ovan, men som använder SIOINUT i PROM för att läsa och skriva i kanalerna A och B.

5. SIOUART

Assebler program för styrning av SIO och CTC såväl som ett UART-kort med interrupt.
Skriven för att användas från prommerbar BASIC.
Läsrutin för kanal A, skrivrutin för kanal B och läsrutin för ett UART kort.

6. AVBROTT

BASIC program (Prommerbart)
Initiera CTC och SIO.
Använder SIOUART rutinerna.
Läser rader både från UART och SIO-kanal A. Sänder raderna till SIO-kanal B.

```
PLC     B   KOD     ARG     LC   STATEMENT----------------------------------
0000                         1   CETEST    PROG  ===== HOW TO USE SIO IN ASYNCRON MODE =====
                             2   *
                             3   *
                             4   *
                             5   *         CE    79-03-05
                             6   *
                             7   *
                             8   *
0000                         9             RADIX 16
0000                        10             EJECT
```

USE THE SIO AS UART (WITHOUT INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD       ARG.    LC   STATEMENT--------------------------------
0000                          11          TITLE USE THE SIO AS UART (WITHOUT INTERUPT)
                              12   *
                              13   *
6000                          14          ORG   06000H
                              15   *
6000                  6000    16   BEGIN  EQU   *
6000        F3                17          DI                      MAKE SURE THAT INTERUPT IS OFF
6001        312263            18          LA    SP,STACK
                              19   *
6004        AF                20          XR    A                 WE MUST FIRST CLEAR THE SIO, BOTH CHANNE
6005        D339              21          OUT   SIOA.CMD          AND DO THAT BY SELECTING  REG 0 FIRST
6007        D33B              22          OUT   SIOB.CMD
6009        3E18              23          LI    A,3*8H            AND THEN COMMAND 3 WHICH IS CHANNEL RESE
600B        D339              24          OUT   SIOA.CMD
600D        D33B              25          OUT   SIOB.CMD
                              26   *
600F        0605              27          LI    B,5               WAIT FOR THE SIO TO COOL DOWN
6011                  6011    28   DELAY  EQU   *
6011        10FE              29          DJNZ  DELAY
                              30   *
6013        013906            31          LA    BC,6:SIOA.CMD     NOW IT'S TIME TO SET UP THE SIO
6016        215D60            32          LA    HL,SIOA.BLK       HL-> ON A BLOCK CONTAINING THE PARAMETER
6019        EDB3              33          BOUT  IR                6 BYTES TO THE SIO
601B        013202            34          LA    BC,2:CTC2         THEN WE MUST USE THE CTC 2
601E        EDB3              35          BOUT  IR                AS BAUD GENERATOR
6020                          36          EJECT
```

USE THE SIO AS UART (WITHOUT INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD     ARG     LC   STATEMENT----------------------------------
                            37   *
                            38   *
                            39   *        LITTLE TEST PROGRAM FOR THE SIO
                            40   *
                            41   *
6020                 6020   42   MAIN     EQU   *
6020        DB06            43            IN    DOG          HIT DOG WHILE WAITING
6022        0E39            44            LI    C,SIOA.CMD    MAKE SURE THAT THE CHANNEL NR IS IN C
6024        CD2C60          45            CALL  READCHAR     CHECK IF ANY CHAR THERE
6027        DC4060          46            CC    SENDSIO       SEND IT AWAY IF SO
602A        18F4            47            JMPS  MAIN
602C                        48            EJECT
```

USE THE SIO AS UART (WITHOUT INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD     ARG     LC   STATEMENT----------------------------------
                            49   *
                            50   *          READ A CHAR FROM SIO
                            51   *
                            52   *          ENTRY: C= PORT CMD ADRESS
                            53   *          EXIT:  CY=1 IF CHAR AND A=CHAR
                            54   *                 CY=0 IF NO CHAR THERE
                            55   *
602C                 602C   56   READCHAR   EQU   *
602C        AF              57              XR    A                 FIRST SELECT SIO REG 0
602D        ED79            58              OUTR  A
602F        ED78            59              INPR  A                 THEN READ REG 0 AND CHECK IF ANY CHAR TH
6031        0F              60              RRCA
6032        D0              61              RNC                     NO THERE
6033        3E01            62              LI    A,1               SOMETHING IS THERE,SELECT SIO REG 1
6035        ED79            63              OUTR  A                 TO FIND OUT IF ANY ERROR WITH THE CHAR
6037        ED78            64              INPR  A
6039        E670            65              NI    070H              MASK UNSIGNIFICANT BITS
603B        2006            66              JNZS  SOMEERR
                            67   *                                  CHAR RECEIVED OK
603D        0D              68              DECR  C                 GET THE DATA PORT NUMBER
603E        ED78            69              INPR  A                 READ THE CHAR
6040        0C              70              INCR  C
6041        37              71              STC
6042        C9              72              RET
                            73   *
                            74   *
                            75   *
6043                 6043   76   SOMEERR    EQU   *                 IF SOME ERROR
6043        0D              77              DECR  C                 DO A DUMMY READ
6044        ED78            78              INPR  A
6046        0C              79              INCR  C
6047        3E30            80              LI    A,6*8             AND THEN IF WE WANT  WE CAN RESET THE
6049        ED79            81              OUTR  A                 ERROR BITS
604B        B7              82              OR    A                 INDICATE NO CHAR
604C        C9              83              RET
604D                        84              EJECT
```

```
PLC     B   KOD     ARG     LC   STATEMENT----------------------------------
                            85   *
                            86   *        SEND A CHAR ON SIO
                            87   *
                            88   *        ENTRY: C= PORT CMD ADRESS
                            89   *                A= CHARACTER TO SENT
                            90   *
604D                604D    91   SENDSIO  EQU   *
604D        F5              92            PUSH  A
604E        AF              93            XR    A              SELECT READ REG 0
604F        ED79            94            OUTR  A
6051                6051    95   WAIT1    EQU   *
6051        ED78            96            INPR  A              IS IT POSSIBLE TO SEND CHAR
6053        CB57            97            BIT   2,A
6055        28FA            98            JZS   WAIT1          NO !! WE MUST WAIT
6057        F1              99            POP   A              GET CHAR
6058        0D             100            DECR  C
6059        ED79           101            OUTR  A
605B        0C             102            INCR  C
605C        C9             103            RET
605D                       104            EJECT
```

USE THE SIO AS UART (WITHOUT INTERUPT) P2:CFSIOTST.TST

```
PLC     B   KOD     ARG     LC   STATEMENT----------------------------------
                            105  *
                            106  *          PARAMETERS TO THE SIO IN ASYNCRON
                            107  *          MODE WITHOUT INTERUPT
                            108  *
605D                 605D   109  SIOA.BLK   EQU    *
605D        03C1            110              DB     3,0C1H          REG 3,8 BIT WORDLENGTH,RECEIVER ENABLE
605F        0444            111              DB     4,044H          REG 4,X 16 CLOCK,1 STOP BIT,NO PARITY
6061        0568            112              DB     5,068H          REG 5,8 BIT WORDLENGTH,TRANSMITTER ENABLE
                            113  *
6063        17              114              DB     017H            CTRL WORD TO CTC
6064        01              115              DB     001H            TIME CONSTANT (X 16 IN SIO AND THIS
                            116  *                                  TIME CONSTANT GIVES 9600 BAUD)
6065                 0039   117  SIOA.CMD    EQU    71Q
6065                 0038   118  SIOA.DAT    EQU    70Q
6065                 003B   119  SIOB.CMD    EQU    73Q
6065                 003A   120  SIOB.DAT    EQU    72Q
6065                 0032   121  CTC2        EQU    62Q
6065                 0006   122  DOG         EQU    6
6065                        123              EJECT
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD       ARG     LC    STATEMENT----------------------------------
6065                          124           TITLE USE THE SIO AS UART (WITH INTERUPT)
                              125   *
                              126   *
                              127   *
6100                          128           ORG   06100H
6100                  6100    129   START   EQU   *
6100          1821            130           JMPS  START1
                              131   *
6110                          132           ALIGN 16                THE INTERUPT VECTORS FOR THE SIO
6110                  6110    133   IVEC    EQU   *
6110          00042061        134           DMA   4,EIRTI           MUST BE ALIGN 16,FIRST WE HAVE VECTORS
                              135   *                               FOR B CHANNEL
                              136   *                               AND THEN FOR A CHANNEL
6118          8961            137           DA    TRANSMIT
611A          2061            138           DA    EIRTI
611C          7A61            139           DA    RECEIVE
611E          2061            140           DA    EIRTI
                              141   *
                              142   *
                              143   *
6120                  6120    144   EIRTI   EQU   *
6120          FB              145           EI
6121                  6121    146   RTI     EQU   *
6121          ED4D            147           RTI
                              148   *
6123                          149           EJECT
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD     ARG     LC   STATEMENT----------------------------------
6123            6123    153  START1   EQU   *
6123        F3          151           DI                    INTERUPT OFF
6124        ED5E        152           IM    2               SELECT INTERUPT MODE 2
6126        312263      153           LA    SP,STACK
                        154  *
6129        0606        155           LI    B,6             MAKE SURE THAT DAISY CHAIN IS CLEARED
612B            612B    156  CLDAISY  EQU   *
612B        CD2161      157           CALL  RTI
612E        10FB        158           DJNZ  CLDAISY
                        159  *
6130        210061      160           LA    HL,START
6133        7C          161           LR    A,H             LOAD THE HIGH ADRESS OF THE
6134        ED47        162           LR    I,A             INTERUPT VECTOR
                        163  *
6136        AF          164           XR    A               WE MUST FIRST CLEAR THE SIO, BOTH CHANNEL
6137        D339        165           OUT   SIOA.CMD        AND DO THAT BY SELECTING REG 0 FIRST
6139        D33B        166           OUT   SIOB.CMD
613B        3E18        167           LI    A,3*8H          AND THEN COMMAND 3 WHICH IS CHANNEL RESET
613D        D339        168           OUT   SIOA.CMD
613F        D33B        169           OUT   SIOB.CMD
                        170  *
6141        0605        171           LI    B,5             WAIT FOR THE SIO TO COOL DOWN
6143            6143    172  DELAY1   EQU   *
6143        10FE        173           DJNZ  DELAY1
                        174  *
                        175  *                             INITIALIZE THE SIO
                        176  *
6145        013804      177           LA    BC,4:SIOB.CMD   IF WE WANT TO USE A CHANNEL ONLY
6148        21A761      178           LA    HL,SIOB.BLK     WE MUST INIT THE B CHANNEL ALSO
614B        EDB3        179           BOUT  IR              IT IS THE INTERUPT VECTOR AND STATUS AFFI
614D        013908      180           LA    BC,8:SIOA.CMD   VECTOR BIT WHICH IS ONLY IN B CHANNEL
6150        EDB3        181           BOUT  IR              SET UP SIO A
6152        013202      182           LA    BC,2:CTC2       USE CTC 2 AS BAUD GENERATOR
6155        EDB3        183           BOUT  IR
                        184  *
                        185  *
                        186  *
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD      ARG     LC   STATEMENT----------------------------------
6157         AF               187          XR    A
6158         D338             188          OUT   SIOA.DAT        DO A DUMMY OUTPUT SO THE INTERUPT
                              189  *                              FOR TRANSMITTER STARTS
615A         21B561           190          LA    HL,STRING
615D         220263           191          STD   HL,STRPNT       USED IN TRANSMITER INTERUPT ROUTINE
6160         FB               192          EI                    TURN ON MAIN SWITCH
6161                          193          EJECT
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD      ARG     LC   STATEMENT----------------------------------
                              194  *
                              195  *          MAIN TEST PROGRAM
                              196  *
6161                  6161    197  LOOP1      EQU    *
6161        DB06              198             IN     DOG              HIT DOG WHILE WAITING
6163        3A0163            199             L      A,RECFLG         CHECK IF ANY CHAR THERE
6166        B7                200             OR     A
6167        28F8              201             JZS    LOOP1
6169        3A0063            202             L      A,CHAR           GET THE CHARACTER WHICH WAS RECEIVED
616C        47                203             LR     B,A              ON THE SIO
616D        3E3D              204             LI     A,75Q            AND SEND IT AWAY ON A USUALI4680 UART CA
616F        D301              205             OUT    CS
6171        78                206             LR     A,B              MUST BE 9600 BAUD TOO
6172        D300              207             OUT    DATA
6174        AF                208             XR     A
6175        320163            209             ST     A,RECFLG
6178        18E7              210             JMPS   LOOP1
617A                          211             EJECT
```

USE THE SIO AS UART (WITH INTERUPT) P2:CFSIOTST.TST

```
PLC     B   KOD      ARG     LC   STATEMENT----------------------------------
                              212  *
                              213  *          INTERUPT ROUTINE FOR RECEIVER
                              214  *
617A                 617A     215  RECEIVE    EQU   *
617A         08               216             EXR   A
617B         DB38             217             IN    SIOA.DAT
617D         320063           218             ST    A,CHAR
6180         3EFF             219             LI    A,0FFH          SET FLAG TO INDICATE THAT WE HAVE ONE CH
6182         320163           220             ST    A,RECFLG
6185         08               221             EXR   A
6186         FB               222             EI
6187         ED4D             223             RTI                   RETURN FROM INTERUPT
6189                          224             EJECT
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD       ARG     LC   STATEMENT--------------------------------
                              225  *
                              226  *          INTERUPT ROUTINE FOR TRANSMITTER
                              227  *
6189                  6189    228  TRANSMIT   EQU    *
6189        D9                229             EXR    R
618A        08                230             EXR    A
618B        ED5B0263          231             LD     DE,STRPNT        GET STRING POINTER
618F        21BF61            232             LA     HL,STREND        CHECK IF AT END
6192        B7                233             OR     A                CY=0
6193        ED52              234             SBDR   HL,DE
6195        2003              235             JNZS   NOTEND
6197        11B561            236             LA     DE,STRING
619A                  619A    237  NOTEND     EQU    *
619A        1A                238             L      A,(DE)
619B        D338              239             OUT    SIOA.DAT         SEND THE CHARACTER
619D        13                240             INCD   DE               UPDATE POINTER
619E        ED530263          241             STD    DE,STRPNT
61A2        D9                242             EXR    R
61A3        08                243             EXR    A
61A4        FB                244             EI
61A5        ED4D              245             RTI
61A7                          246             EJECT
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

```
PLC     B   KOD       ARG     LC   STATEMENT----------------------------------
                              247  *
                              248  *          PARAMTERS FOR SIO WHEN USING INTERUPT
                              249  *
61A7                  61A7    250  SIOB.BLK   EQU    *
61A7        0210              251             DB     2,IVEC          REG 2 IN B CHANNEL IS INT VECTORS
61A9        0104              252             DB     1,004H          STATUS AFFECTS VECTOR BIT IS ONLY IN B
                              253  *
61AB        011A              254             DB     1,01AH          REG 1,INT ON ALL CHARS,TRANSM INT ON
61AD        03C1              255             DB     3,0C1H          REG 3,8 BIT WORDLENGTH,RECEIVER ENABLE
61AF        0444              256             DB     4,044H          REG 4,X 16 CLOCK,1 STOP BIT,NO PARITY
61B1        0568              257             DB     5,068H          REG 5,8 BIT WORDLENGTH,TRANSMITTER ENABLE
                              258  *
61B3        17                259             DB     017H            CTRL WORD TO CTC
61B4        01                260             DB     001H            TIME CONSTANT (X 16 IN SIO AND THIS
                              261  *                                  TIME CONSTANT GIVES 9600 BAUD)
                              262  *
61B5                  61B5    263  STRING     EQU    *
61B5        54455354          264             DB     'TESTING',15Q,12Q
61BE                  61BE    265  STRFND     EQU    *
                              266  *
61BE                  0000    267  DATA       EQU    0
61BE                  0001    268  CS         EQU    1
                              269  *
0000                          270             PLC    1
6300                          271             ORG    06300H          RAM AREA
6300                          272  CHAR       DS     1
6301                          273  RECFLG     DS     1
6302                          274  STRPNT     DS     2
6304                          275             DS     30
6322                  6322    276  STACK      EQU    *
6322                          277             END    BEGIN
ERRORS:    0    WARNINGS:    0
```

USE THE SIO AS UART (WITH INTERUPT) P2:CESIOTST.TST

PLC	B	KOD	ARG	LC	STATEMENT					
BEGIN		6000	16	277						
CHAR		6300	272	202	218					
CLDAISY		612B	156	158						
CS		0001	268	205						
CTC2		0032	121	34	182					
DATA		0000	267	207						
DELAY		6011	28	29						
DELAY1		6143	172	173						
DOG		0006	122	43	198					
EIRTI		6120	144	134	138	140				
IVEC		6110	133	251						
LOOP1		6161	197	201	210					
MAIN		6020	42	47						
NOTEND		619A	237	235						
READCHAR		602C	56	45						
RECEIVE		617A	215	139						
RECFLG		6301	273	199	209	220				
RTI		6121	146	157						
SENDSIO		604D	91	46						
SIOA.BLK		605D	109	32						
SIOA.CMD		0039	117	21	24	31	44	165	168	180
SIOA.DAT		0038	118	186	217	239				
SIOB.BLK		61A7	250	178						
SIOB.CMD		003B	119	22	25	166	169	177		
SIOB.DAT		003A	120							
SOMEERR		6043	76	66						
STACK		6322	276	18	153					
START		6100	129	160						
START1		6123	150	130						
STREND		61BE	265	232						
STRING		61B5	263	190	236					
STRPNT		6302	274	191	231	241				
TRANSMIT		6189	228	137						
WAIT1		6051	95	98						

```
1000 REM Sio1
1010 REM För att kunna överföra tecken med hjälp av SIO kretsen krävs att den
1020 REM initieras. Dessutom måste en CTC kanal användas till baudrate generering.
1030 REM När man ska överföra tecken med flaggtest krävs att skrivregister 0,3,4,5
1040 REM initieras på SIO kretsen. För att initiera kretsen finns det en kommandoport
1050 REM till varje kanal. Initieringen utförs i två steg. Först väljer man det
1060 REM register som  ska initieras. Sedan skickar man på initieringsdata.
1070 REM ---------------------------------------------------------------------
1080 REM Initiering av SIO kanal A.
1090 REM
1100 REM Vi börjar med register 0.
1110 REM
1120 REM Välj in register 0 skicka 24 som står för RESET.
1130 REM A-kanalens kommando port=57
1140 REM A-kanalens data port=56
1150 REM
1160 REM
1170 OUT 57,0,57,24
1180 REM
1190 REM Register 3 innehåller information som har att göra med mottagning av
1200 REM tecken.
1210 REM
1220 REM '  7  6  5  4  3  2  1  0.        Bitnr.
1230 REM ' ------------------------
1240 REM '  1  1  x  x  x  x  x  1         x ointressant (sätts lika med 0)
1250 REM '                     -!---tillåt mottagning
1260 REM
1270 REM ' -!--!---------------------8 bitar per tecken
1280 REM
1290 OUT 57,3,57,128+64+1
1300 REM
1310 REM I register 4 kan man välja antalet stoppbitar och hastigheten på
1320 REM baudrate generatorn
1330 REM
1340 REM '  7  6  5  4  3  2  1  0
1350 REM ' ------------------------
1360 REM '  0  1  x  x  0  1  x  x
1370 REM '              -!--!--------1 stopp bit
1380 REM ' -!--!--------------------baudrate generatorn går 16 ggr fortare än
1390 REM '                          aktuell baudrate.
1400 REM
1410 OUT 57,4,57,64+4
1420 REM
1430 REM
1440 REM
```

```
1450 REM
1460 REM
1470 REM
1480 REM
1490 REM I register 5 styr man sändningen.
1500 REM
1510 REM '  7  6  5  4  3  2  1  0
1520 REM ' ------------------------
1530 REM '  x  1  1  x  1  x  x  x
1540 REM '              -!-----------tillåter sändning
1550 REM '     -!--!-----------------8 bitar per tecken
1560 REM
1570 OUT 57,5,57,64+32+8
1580 REM ----------------------------------------------------------------------
1590 REM Initiering av CTC
1600 REM
1610 REM
1620 REM Vi väljer CTC2 som baudrate generator.
1630 REM CTC2 har port nr. 50
1640 REM
1650 REM
1660 REM '  7  6  5  4  3  2  1  0
1670 REM ' ------------------------
1680 REM '  0  0  0  x  x  1  1  1
1690 REM '                      -!--kontroll ord
1700 REM '                   -!-----reset
1710 REM '                -!--------anger att tidskonstant följer
1720 REM '        -!-----------------prescaler 16
1730 REM '     -!--------------------timer mode
1740 REM ' -!------------------------ej avbrott
1750 REM
1760 OUT 50,7
1770 REM
1780 REM Skriv direkt ner en tidskonstant
1790 REM
1800 OUT 50,1
1810 REM
1820 REM Tidskonstanten 1 och prescaler 16 samt systemklockan på 2.5 MHZ ger
1830 REM en frekvens på 1/(1*16*400E-9)=1.56E5.
1840 REM Vi hade valt 16 ggr baudrate som ger 16*9600=1.54E5. Avvikelsen är
1850 REM acceptabel.
1860 REM
1870 REM
1880 REM
1890 REM
1900 REM
```

```
1910 REM
1920 REM
1930 REM
1940 REM ---------------------------------------------------------------------
1950 REM Vi skriver ett enkelt program som läser tecken från A kanalen tills
1960 REM ett CR kommer in.
1970 REM Sedan ekas alla tecken ut på A kanalen.
1980 REM
1990 REM Läs in en rad till strängen A$
2000 GOSUB 2080
2010 REM Lägg till ett LF och eka strängen
2020 A$=A$+CHR$(10)
2030 GOSUB 2230
2040 GOTO 2000
2050 REM ---------------------------------------------------------------------
2060 REM Läsrutin
2070 REM
2080 REM För att ta reda på om tecken har kommit in läser man av SIO kretsens
2090 REM läs register 0
2100 A$=''
2110 REM Börja med att välja register 0
2120 OUT 57,0
2130 S=INP(57)
2140 REM Om bit 0 (med värdet 1) är satt finns det tecken att hämta
2150 IF (S AND 1) THEN T1=INP(56) ELSE GOTO 2120
2160 REM Maska bort eventuell paritet
2170 T1=T1 AND 127
2180 A$=A$+CHR$(T1)
2190 IF T1=13 THEN RETURN ELSE GOTO 2120
2200 REM ---------------------------------------------------------------------
2210 REM Skriv rutin
2220 REM
2230 FOR I=1 TO LEN(A$)
2240   OUT 57,0
2250   S=INP(57)
2260   REM Om bit 2 (med värdet 4) är satt kan man skicka nästa tecken.
2270   IF (S AND 4) THEN GOTO 2280 ELSE GOTO 2240
2280   T1=ASCII(MID$(A$,I,1))
2290   OUT 56,T1
2300 NEXT I
2310 RETURN
```

```
SIOINUT   PROG  LÄS FRÅN SIO-A / SKRIV TILL SIO-B
*
*         VID ANROP PEKAR DE PÅ EN STRÄNG DÄR POSITION 1 ANGER MAX
*         ANTAL TECKEN SOM FÅR MOTTAGAS. I DE EFTERFÖLJANDE POSITIONERNA
*         SKA TECKNEN LAGRAS.
*
*         -------------------------------
*         ! N !   PLATS FÖR N ST TECKEN !
*         -------------------------------
*
*
          L     A,(DE)          DE PEKAR PÅ PARAMETER I BASIC ANROPET:
*                               A=CALL(ADRESS,PARAMETER)
          LR    B,A             FLYTTA MAX LÄNGDEN TILL REGISTER B
          PUSH  DE
          POP   HL              LÅT HL PEKA PÅ BYTE NR 1
          INCD  DE              DE PEKAR PÅ FÖRSTA TECKNET
          LI    C,57            SIO-A KOMMANDO PORT
LMOT      =     *
          INP   6               SLÅ TILL 'WATCHDOGEN'
          XR    A
          OUTR  A               VÄLJ REGISTER 0
          INPR  A               LÄS REGISTER 0
          NI    1               BIT 0 ANGER OM TECKEN FINNS
          JZS   LMOT            OM BIT 0 = 0 FINNS DET INGET TECKEN
          DECR  C               C SIO-A DATA PORT
          INPR  A               LÄS TECKEN
          NI    127             MASKA BORT PARITET
          ST    A,(DE)          LAGRA TECKEN
          INCD  DE
          INCR  C               C SIO-A KOMMANDO PORT IGEN
          CI    13              TERMINERA PÅ CR
          JZS   TERM
          DJNZ  LMOT            TERMINERA PÅ MAX ANTAL TECKEN OCKSÅ
          RET                   MAX ANTAL TECKEN STÅR REDAN I POS. 1
TERM      =     *
          L     A,(HL)
          SR    B               BERÄKNA VERKLIGT ANTAL TECKEN SOM
          ST    A,(HL)          HAR MOTTAGITS. LAGRA I POSITION 1.
          RET
*
          EJECT
*
*         SÄND TECKEN TILL SIO KANAL B
*         ANROPAS FRÅN BASIC ENLIGT SAMMA FORMAT SOM OVAN.
```

```
*
        L       A,(DE)          HHÄMTA ANTAL TECKEN
        LR      B,A
        INCD    DE
        LI      C,59            SIO-B KOMMANDO PORT
LSÄND   =       *
        INP     6               SLÅ TILL 'WATCHDOGEN'
        XR      A               A=0
        OUTR    A               VÄLJ REGISTER 0
        INPR    A               LÄS REGISTER 0
        NI      4               OM BIT 2 = 1 GÅR DET ATT SKICKA NÄSTA TECKEN
        JZS     LSÄND
        L       A,(DE)
        INCD    DE              HÄMTA NÄSTA TECKEN OCH UPPDATERA PEKARE
        DECR    C               SIO-B DATA PORT
        OUTR    A               SKICKA TECKNET
        INCR    C               SIO-B KOMMANDO PORT IGEN
        DJNZ    LSÄND
        RET
*
        END
```

```
1000 REM SIO2
1010 REM I det här exemplet har vi lagt två stycken assembly rutiner i PROM som
1020 REM anropas från ett BASIC program. En assembly rutin läser från kanal A.
1030 REM En annan assembly rutin skriver på kanal B. Initieringen av sio kanalerna
1040 REM går till på samma sätt som i exempel Sio1.
1050 REM
1060 REM Initiering av sio A
1070 REM -------------------
1080 OUT 57,0,57,24
1090 OUT 57,3,57,128+64+1
1100 OUT 57,4,57,64+4
1110 OUT 57,5,57,64+32+8
1120 OUT 50,7,50,1 : REM CTC2
1130 REM
1140 REM Initiering av sio B
1150 REM -------------------
1160 OUT 59,0,59,24
1170 OUT 59,3,59,128+64+1
1180 OUT 59,4,59,64+4
1190 OUT 59,5,59,64+32+8
1200 OUT 49,7,49,1 : REM CTC1 används till baudrate generator.(9600 baud)
1210 REM
1220 REM ---------------------------------------------------------------------
1230 REM Ett litet program som använder sig av två assembly rutiner som är
1240 REM lagda i PROM. SE SIOINUT.
1250 REM
1260 POKE -128,80 : REM Max antal tecken skrivs ner i position 1
1270 A=CALL(14336,-128) : REM läs tecken från sio A
1280 A=CALL(14373,-128) : REM skriv tecknen på sio B
1290 GOTO 1260
```

```
SIOUART   PROG  AVBROTTSRUTINER FÖR SIO,UART
*
          RADIX 16
          ORG   3800H
          LIST  IF
ABC80     =     1
ABC800    =     1-ABC80
*
*
*         TABELL SOM PEKAR UT DE OLIKA AVBROTTS RUTINERNA
*         DE FÖRSTA FYRA POSITIONERNA GÄLLER CTC DÄR NOLLAN
*         OCH TVÅAN INTE GENERERAR AVBROTT.
*         DE POSITIONER SOM PEKAR PÅ RETI ANVÄNDS INTE.
*
          IF    ABC80
KLOCKA    =     59
POKEAREA  =     -128
          ENDF
*
          IF    ABC800
KLOCKA    =     47
POKEAREA  =     -512
          ENDF
*
INTTAB    =     *
*
*         SIO KANAL B
*
          DA    SÄNDB              RUTIN FÖR SÄNDNING KANAL B
          DA    RETI
          DA    RETI
          DA    RETI
*
*         SIO KANAL A
*
          DA    RETI
          DA    RETI
          DA    MOTA               RUTIN FÖR MOTTAGNING KANAL A
          DA    RETI
*
CTCVEK    =     *
          DA    RETI
          DA    RETI
          DA    RETI
          DA    KLOCKA
*
```

```
          STRUC
          ORG    POKEAREA
URÄKNA    DS     1
USTATUS   DS     1
UPEKA     DS     2
ARÄKNA    DS     1
ASTATUS   DS     1
APEKA     DS     2
BRÄKNA    DS     1
BSTATUS   DS     1
BPEKA     DS     2
          ENDS
*
*
*
*         INITRUTIN FÖR I-REG OCH AVBROTTSVEKTOR
*
          DI
          LA     HL,INTTAB
          LR     A,H
          LR     I,A                HÖGDELEN SKA LIGGA I I-REGISTRET
          LI     A,2                VÄLJ IN REGISTER 2
          OUT    59                 PÅ SIO-B
          LR     A,L
          OUT    59                 LADDA LÅGDEL AV INTTAB
          LI     A,CTCVEK
          OUT    48                 INITIERA ÄVEN CTC
          EI
          RET
*
          EJECT
*
*         AVBROTTSRUTIN FÖR MOTTAGNING AV TECKEN FRÅN 4017,4117
*
UART      =      *
          PUSH   HL                 PUSHA ALLA REGISTER SOM ANVÄNDS
          PUSH   A
          LI     A,75Q
          OUT    1                  GÖT KORTVAL
          INP    0                  LÄS TECKNET
          NI     127
          OUT    2                  KVITTERA TECKNET
          LD     HL,UPEKA
          ST     A,(HL)
          INCD   HL
          ST     A,USTATUS          SIST MOTTAGNA TECKEN I USTATUS
```

```
        L       A,URÄKNA
        NR      A                       HECKA O BUFFERTEN ÄR FULL
        JZS     UFULL
        DECR    A                       MINSKA RÄKNAREN MED ETT OCH
        ST      A,URÄKNA                LAGRA NYA VÄRDEN PÅ RÄKNARE OCH PÅ...
        STD     HL,UPEKA                ...PEKARE
UFULL   =       *
        POP     A
        POP     HL
RETI    EI
        RTI
*
        EJECT
*
*       AVBROTTSRUTIN SIO-A MOTTAGNING
*
MOTA    =       *
        PUSH    HL
        PUSH    A
        INP     56
        NI      127                     MASKA PARITET
        LD      HL,APEKA
        ST      A,(HL)
        INCD    HL
        ST      A,ASTATUS
        L       A,ARÄKNA                CHECKA OM BUFFERTEN ÄR FULL
        NR      A
        JZS     FULL
        DECR    A
        ST      A,ARÄKNA                MINSKA RÄKNAREN MED ETT
        STD     HL,APEKA                LAGRA PEKAREN TILL NÄSTA TECKEN
FULL    =       *
        POP     A
        POP     HL                      HÄMTA TILLBAKA GAMLA REGISTER VÄRDEN.
        EI
        RTI
*
        EJECT
*
*       AVBROTTSRUTIN FÖR SÄNDNING SIO-B
*
SÄNDB   =       *
        PUSH    HL
        PUSH    A
        L       A,BRÄKNA
        NR      A
```

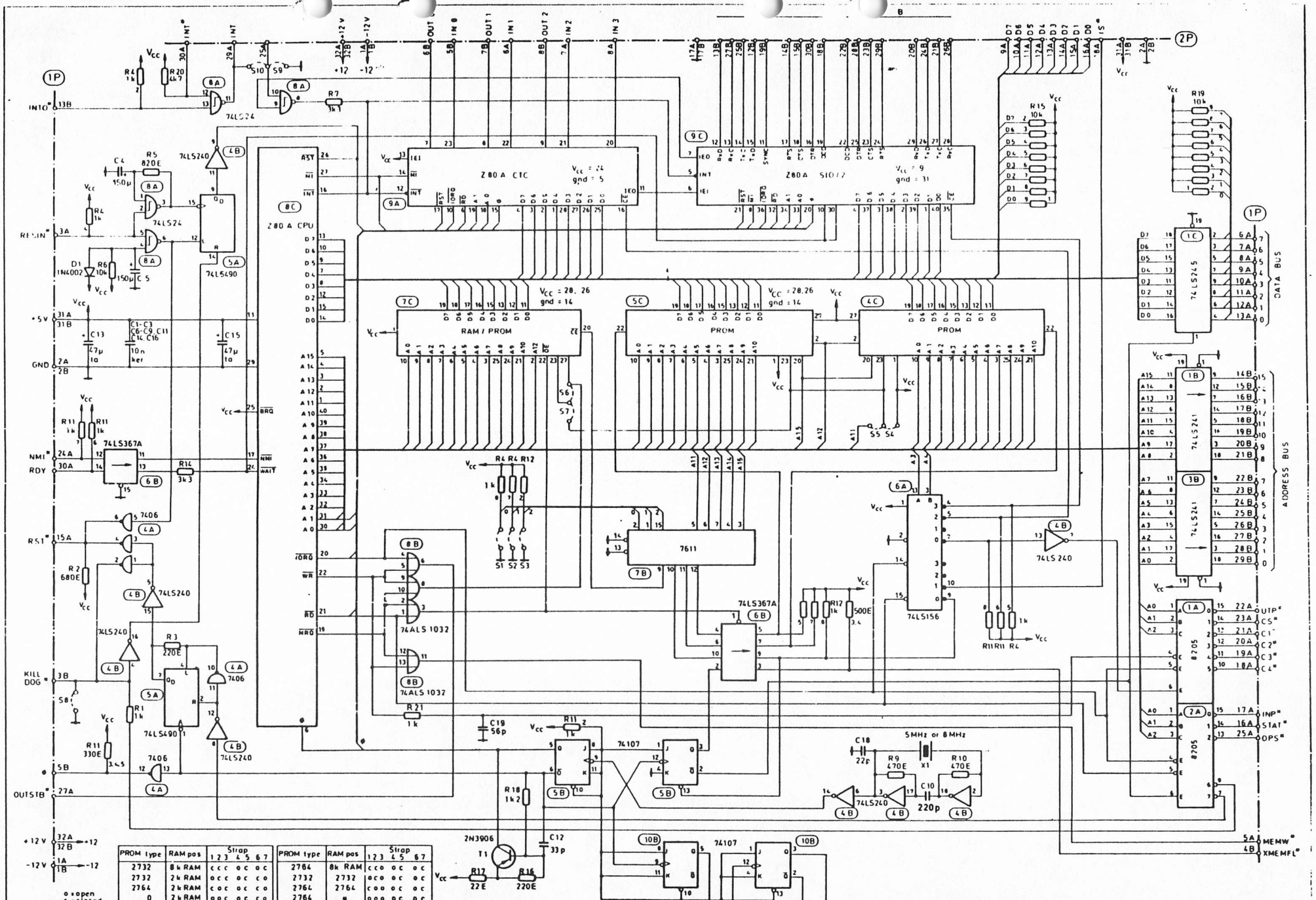
```
        JNZS  EJSLUT
        LI    A,0          OM DET INTE FINNS NAGRA FLER TECKEN
        OUT   59           ATT SÄNDA SA VÄLJER VI IN REGISTER 0
        LI    A,32+8       PA SIO B OCH SLAR IFRAN SÄNDNINGS
        OUT   59           AVBROTTET.
        LI    A,0
        ST    A,BSTATUS    INDIKERA ATT ÖVERFÖRINGEN ÄR KLAR
        JMPS  RET
*
EJSLUT  =     *
        DECR  A
        ST    A,BRÄKNA     RÄKNA NER TECKEN RÄKNAREN.
        LD    HL,BPEKA     HÄMTA PEKAREN TILL NÄSTA TECKEN.
        L     A,(HL)       HÄMTA TECKNET OCH...
        INCD  HL
        STD   HL,BPEKA
        OUT   58           ...SÄND IVÄG TECKNET TILL SIO-B.
RET     =     *
        POP   A
        POP   HL
        EI
        RTI
*
        ORG   38FFH        EXTERNT AVBROTT GET AVBROTTSVEKTOR
        DA    UART         I-REG+0FFH. I DETTA FALL 38FFH
        END
```

```
1000 REM Avbrott
1010 REM I det här exemplet ska sio kretsen användas med avbrottshantering.
1020 REM Data kommer in från sio A samt från ett uart kort 4017/4117. Inläsningen
1030 REM går med avbrottshantering. All data som kommer in skicka vidare till sio
1040 REM B som också jobbar med avbrott. Allt detta administreras från ett
1050 REM BASIC program.
1060 REM ----------------------------------------------------------------------
1070 REM Initiering av SIO kanal A.
1080 REM
1090 REM Vi börjar med register 0.
1100 REM
1110 REM Välj in register 0 skicka 24 som står för RESET.
1120 REM A-kanalens kommando port=57
1130 REM A-kanalens data port=56
1140 REM
1150 REM
1160 OUT 57,0,57,24
1170 REM
1180 REM Register 1 innehåller information som har att göra med avbrottshantering.
1190 REM
1200 REM '  7  6  5  4  3  2  1  0
1210 REM ' ------------------------
1220 REM '  x  x  x  1  1  x  x  x
1230 REM '                   -!------Avbrott för sändning(kanal B).
1240 REM '                -!---------Ska sättas i kanal B (se nedan). Påverkar
1250 REM '                            avbrottsvektorn.
1260 REM '           !--!------------Avbrott vid mottagning(pariteten påverkar
1270 REM '                            inte avbrottsvektorn.)
1280 REM
1290 OUT 57,1,57,16+8
1300 REM
1310 REM Register 3 innehåller information som har att göra med mottagning av
1320 REM tecken.
1330 REM
1340 REM '  7  6  5  4  3  2  1  0        Bitnr.
1350 REM ' ------------------------
1360 REM '  1  1  x  x  x  x  x  1        x ointressant (sätts lika med 0)
1370 REM '                      -!---tillåt mottagning
1380 REM
1390 REM ' -!--!---------------------8 bitar per tecken
1400 REM
1410 OUT 57,3,57,128+64+1
1420 REM
1430 REM I register 4 kan man välja antalet stoppbitar och hastigheten på
1440 REM baudrate generatorn
```

```
1450 REM
1460 REM '  7  6  5  4  3  2  1  0
1470 REM ' ------------------------
1480 REM '  0  1  x  x  0  1  x  x
1490 REM '              -!--!--------1 stopp bit
1500 REM ' -!--!--------------------baudrate generatorn går 16 ggr fortare än
1510 REM '                          aktuell baudrate.
1520 REM
1530 OUT 57,4,57,64+4
1540 REM
1550 REM
1560 REM
1570 REM
1580 REM
1590 REM
1600 REM
1610 REM I register 5 styr man sändningen.
1620 REM
1630 REM '  7  6  5  4  3  2  1  0
1640 REM ' ------------------------
1650 REM '  x  1  1  x  1  x  x  x
1660 REM '              -!-----------tillåter sändning
1670 REM '    -!--!-----------------8 bitar per tecken
1680 REM
1690 OUT 57,5,57,64+32+8
1700 REM ---------------------------------------------------------------------
1710 REM Initiering av CTC
1720 REM
1730 REM
1740 REM Vi väljer CTC2 som baudrate generator.
1750 REM CTC2 har port nr. 50
1760 REM
1770 REM
1780 REM '  7  6  5  4  3  2  1  0
1790 REM ' ------------------------
1800 REM '  0  0  0  x  x  1  1  1
1810 REM '                     -!--kontroll ord
1820 REM '                  -!-----reset
1830 REM '               -!--------anger att tidskonstant följer
1840 REM '        -!-----------------prescaler 16
1850 REM '    -!--------------------timer mode
1860 REM ' -!-----------------------ej avbrott
1870 REM
1880 OUT 50,7
1890 REM
1900 REM Skriv direkt ner en tidskonstant
```

```
1910 REM
1920 OUT 50,1
1930 REM
1940 REM Tidskonstanten 1 och prescaler 16 samt systemklockan på 2.5 MHZ ger
1950 REM en frekvens på 1/(1*16*400E-9)=1.56E5.
1960 REM Vi hade valt 16 ggr baudrate som ger 16*9600=1.54E5. Avvikelsen är
1970 REM acceptabel.
1980 REM
1990 REM
2000 REM
2010 REM
2020 REM
2030 REM
2040 REM
2050 REM
2060 REM ---------------------------------------------------------------------
2070 REM Initiering av kanal B
2080 REM ---------------------
2090 OUT 59,0,59,24
2100 OUT 59,1,59,4+2 : REM Se kommentar ovan
2110 REM Register 2 initieras av assembly rutin.  SIOVART
2120 OUT 59,3,59,128+64+1
2130 OUT 59,4,59,64+4
2140 OUT 59,5,59,64+32+8
2150 OUT 49,7,49,1 : REM CTC1 ANVÄNDS SOM BAUDRATEGENERATOR
2160 REM ---------------------------------------------------------------------
2170 REM Initiera pekare och räknare i pokearean
2180 B=-116
2190 POKE -128,20,0,B,SWAP%(B)
2200 POKE -124,20,0,B+20,SWAP%(B+20)
2210 POKE -120,0,0,B+40,SWAP%(B+40)
2220 REM Initiera I reg mm.
2230 A=CALL(14360)
2240 REM Slå på avbrott på UART kortet.
2250 OUT 1,61,5,64
2260 REM ---------------------------------------------------------------------
2270 REM Huvudslinga
2280 REM
2290 REM Rader läses in från uart kort och från
2300 REM sio-a. De ekas på sio-b tills nästa rad
2310 REM kommer in.
2320 REM
2330 REM Checka om det har kommit in en rad på uarten.
2340 IF PEEK(-127)<>13 THEN 2460
2350 REM Beräkna längden på raden.
2360 A=20-PEEK(-128)
```

```
2370 A$=''
2380 REM Läs in raden till A$.
2390 FOR I=0 TO A-1
2400   T=PEEK(B+I)
2410   A$=A$+CHR$(T)
2420 NEXT I
2430 REM Indikera att ny rad kan läsas in
2440 POKE -128,20,0,B,SWAP%(B)
2450 REM Checka om rad har inkommit på SIO-A
2460 IF PEEK(-123)<>13 THEN 2550
2470 A=20-PEEK(-124)
2480 B$=''
2490 FOR I=0 TO A-1
2500   T=PEEK(B+20+I)
2510   B$=B$+CHR$(T)
2520 NEXT I
2530 REM Indikera att ny rad kan läsas in.
2540 POKE -124,20,0,B+20,SWAP%(B+20)
2550 IF A$<>'' THEN C$=A$ : GOSUB 2610
2560 IF B$<>'' THEN C$=B$ : GOSUB 2610
2570 GOTO 2330
2580 REM ----------------------------------------------------------------
2590 REM Subrutin som skriver ut en rad
2600 REM
2610 REM Checka om föregående sändning är klar.
2620 IF PEEK(-119)<>0 THEN 2620
2630 C$=C$+CHR$(10)
2640 REM Skriv ner raden i sändbufferten
2650 FOR I=1 TO LEN(C$)
2660   T=ASCII(MID$(C$,I,1))
2670   POKE B+40+I-1,T
2680 NEXT I
2690 POKE -120,LEN(C$),-1,B+40,SWAP%(B+40)
2700 REM Sänd iväg första tecknet. De kvarvarande tecknen sänds med avbrott.
2710 A=CALL(14450)
2720 RETURN
```



PROM type	RAM pos	Strap 1 2 3 4 5 6 7
2732	8 k RAM	c c c o c o c
2732	2 k RAM	o c c o c o c
2764	2 k RAM	c o c o c o c
0	2 k RAM	o o c o c o c

PROM type	RAM pos	Strap 1 2 3 4 5 6 7
2764	8k RAM	c c o o c o c
2732	2732	o c o o c o c
2764	2764	c o o o c o c
2764	"	o o o o c o c

o = open
c = closed

S8 c ⇒ no W-dog
S8 o ⇒ W-dog on

S9 c, S10 o = bus interrupts disabled
S9 o, S10 c = " —— " —— enabled

* 2764 i adress 24k- 32k

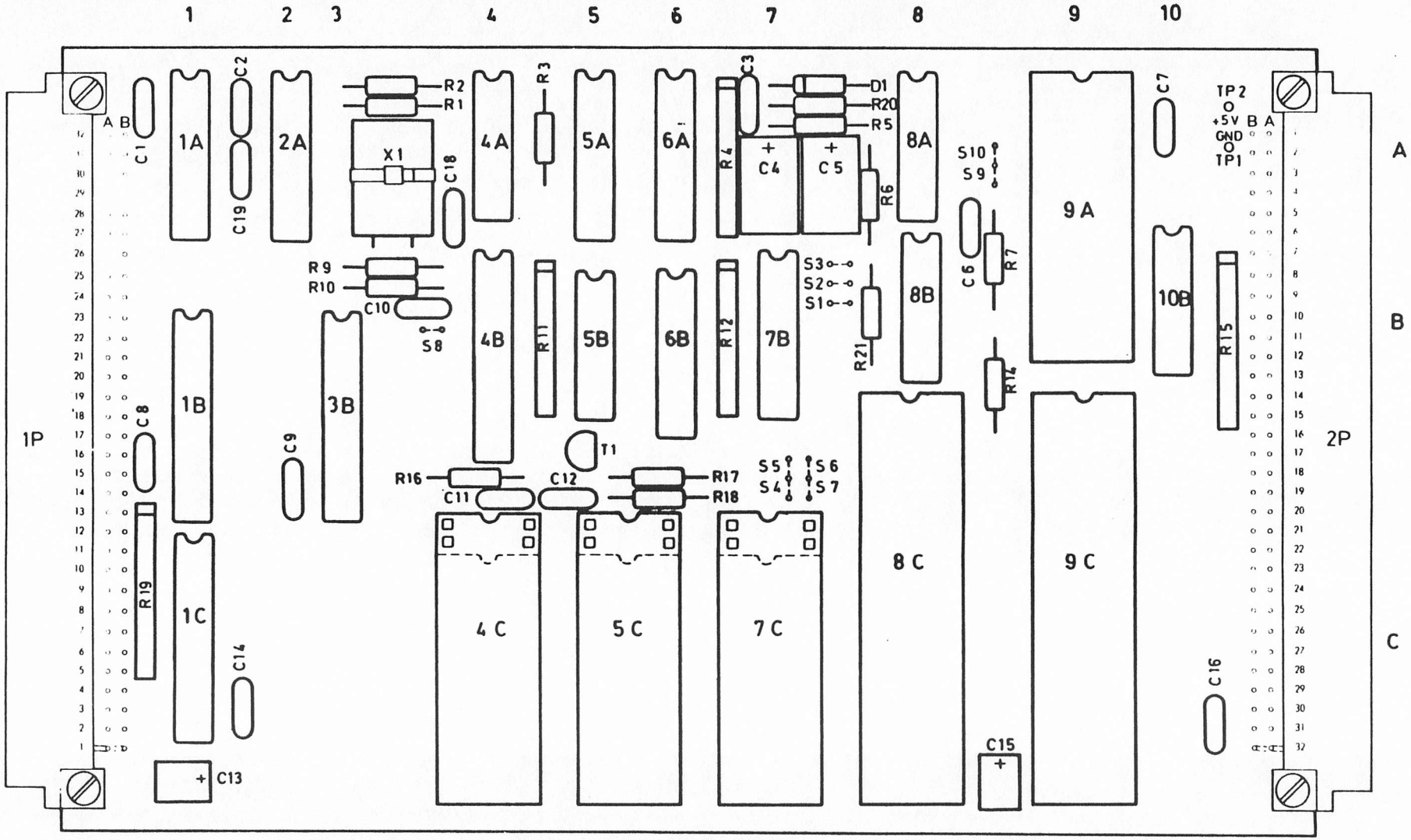
			A	REV	OL
			850702	DATE	840203

Océ 7363-02

DATAINDUSTRIER AB
SWEDEN

SINGLE BOARD COMPUTER SIO

81 - 1004 - 10



PROM type	RAM pos	Strap 1 2 3	Strap 4 5	Strap 6 7
2732	8k RAM	c c c	o c	o c
2732	2k RAM	o c c	o c	c o
2764	2k RAM	c o c	o c	c o
0	2k RAM	o o c	o c	c o

PROM type	RAM pos	Strap 1 2 3	Strap 4 5	Strap 6 7
2764	8k RAM	c c o	o c	o c
2732	2732	o c o	o c	o c
2764	2764	c o o	o c	o c
2764	*	o o o	o c	o c

o = open
c = closed

S8 c ➡ no W-dog
S8 o ➡ W-dog
S9 c S10 o = bus interrupts disabled
S9 o S10 c = " —"— enabled
* 2764 i adress 24k - 32k

				REV	RIT	ÖL	DATAINDUSTRIER AB SWEDEN	SINGLE BOARD COMPUTER SIO	82-1004-10
				DATUM	840203			

DOCUMENTATION COMMENT REPORT Nr: _____

Originator: _____ Date: _____
(Name, Address, _____
Telephone) _____
_____

Manual/Datasheet: _____
_____
Version & Date: _____

Related products _____
and versions: _____
_____

Used in computer: _____
Operating system _____
version: _____
_____
Delivered by: _____

Application: _____
_____
_____

Describe below the deviations found. Give the page, the error and if possible, your suggested correction.

Describe also how the deviation was found and what influence it had on your system.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

PLEASE SEND THIS REPORT TO **DataIndustrier DIAB AB**
Documentation
Box 2029
S-183 02 Täby, Sweden

****

ACTION BY DOCUMENTATION STAFF.

Comment verified Date: _____ Sign: _____

COMMENTS FIXED:

Release notice:	Version _____	Date: _____	Sign: _____
Manual/Datasheet:	Version _____	Date: _____	Sign: _____
Media:	Version _____	Date: _____	Sign: _____
Hardware:	Version _____	Date: _____	Sign: _____

Transferred to distribution: Date: _____ Sign: _____

DataIndustrier DIAB AB